

1-1-2014

Vehicle Lane Departure Prediction Based On Support Vector Machines

Alhadi Ali Albousefi
Wayne State University,

Follow this and additional works at: http://digitalcommons.wayne.edu/oa_dissertations



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Albousefi, Alhadi Ali, "Vehicle Lane Departure Prediction Based On Support Vector Machines" (2014). *Wayne State University Dissertations*. Paper 957.

This Open Access Dissertation is brought to you for free and open access by DigitalCommons@WayneState. It has been accepted for inclusion in Wayne State University Dissertations by an authorized administrator of DigitalCommons@WayneState.

VEHICLE LANE DEPARTURE PREDICTION BASED ON SUPPORT VECTOR MACHINES

by

ALHADI ALI ALBOUSEFI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2014

MAJOR: ELECTRICAL ENGINEERING

Approved by:

Advisor

Date

© COPYRIGHT BY
ALHADI ALI ALBOUSEFI
2014
All Rights Reserved

DEDICATION

I dedicate my humble work to my father who encouraged and supported me to be in the right path, my mother who raised me and still prays for me to be a successful person to people and community, my wife and kids Hammam, Jumana and Abdalmuhymmin, and all my family members and friends who gave me support and help to finish my dissertation.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Professor Hao Ying, who contributed tremendous time to guide my research. Appreciation is also due to Professor Feng Lin, Professor Chandan Reddy, and Professor Caisheng Wang for their constructive comments and valuable suggestions.

TABLE OF CONTENTS

| | |
|---|-----|
| Dedication..... | ii |
| Acknowledgement | iii |
| List of Tables..... | vii |
| List of Figures..... | ix |
| Chapter 1 Introduction | 1 |
| 1.1 Background and Motivation..... | 1 |
| 1.2 Literature Review | 5 |
| <i>1.2.1 Time to Lane Crossing-Based Techniques</i> | 5 |
| <i>1.2.2 Estimating Vehicle Position with Respect to Lane Markings Using Vision Sensors</i> | 8 |
| <i>1.2.3 Estimating Vehicle Position Using Vehicle Variables</i> | 11 |
| 1.3 Research Objectives and My Technical Contributions | 12 |
| 1.4 Introduction to Support Vector Machines | 13 |
| <i>1.4.1 Supervised and Unsupervised Learning</i> | 13 |
| <i>1.4.2 Linear Support Vector Machines</i> | 15 |
| <i>1.4.2.1 Separable classes</i> | 15 |
| <i>1.4.2.2 Non-Separable Classes (Soft Margin Classifier)</i> | 19 |
| <i>1.4.3 Nonlinear Support Vector Machines</i> | 22 |
| <i>1.4.4 Kernel Functions</i> | 24 |

| | |
|---|----|
| 1.4.5 Kernel Selection..... | 25 |
| 1.4.6 Parameter Estimation..... | 26 |
| 1.4.7 Data Preprocessing..... | 26 |
| 1.4.8 Features Selection | 27 |
| Chapter 2 A Two-Stage SVM Training Scheme for Lane Departure Prediction | 28 |
| 2.1 The Two-Stage SVM Training Scheme | 28 |
| 2.2 Experiment Data..... | 32 |
| 2.2.1 Data Source for SVM 1 and SVM2..... | 32 |
| 2.2.2 Description of Variables..... | 34 |
| 2.2.3 Data Analysis..... | 35 |
| 2.2.4 Data Cleaning | 41 |
| 2.3 Settings for SVM Training and Testing | 52 |
| 2.3.1 Preparation of Training Sets A and B..... | 52 |
| 2.3.2 Preparation of Testing Sets A and B | 55 |
| 2.3.3 Other SVM Settings | 56 |
| 2.3.4 Statistical Measures of the Binary Classification Performance..... | 56 |
| 2.3.5 Statistical Pattern Recognition Toolbox for Matlab (STPRtool box)..... | 58 |
| Chapter 3 SVM Lane Departure Prediction Experiment Results | 60 |
| 3.1 SVM Training Results..... | 60 |
| 3.2 SVM Testing Results | 63 |

| | |
|---------------------------------|----|
| Chapter 4 Conclusion..... | 76 |
| References..... | 78 |
| Abstract..... | 85 |
| Autobiographical Statement..... | 87 |

LIST OF TABLES

| | |
|---|----|
| Table 2.1. Illustration of constructing training examples from one lane departure event..... | 54 |
| Table 2.2. Possible classification results of a driver | 57 |
| Table 2.3. Implemented functions for binary support vector machines | 58 |
| Table 2.4. Data type used to represent binary SVMs | 59 |
| Table 3.1. How numbers of false positives and false negatives change with the window size for the RBF kernel SVM 1 using lateral position and lateral velocity as its inputs..... | 61 |
| Table 3.2. Amounts of support vectors and training error for SVM 1 using Training Set A and SVM 2 using Training Set B. Lateral position and lateral velocity are input variables. | 63 |
| Table 3.3. Numbers of false negatives and false positives generated during the testing of the RBF kernel SVM 1 and the second-order polynomial kernel SVM 1..... | 64 |
| Table 3.4. Testing results of RBF SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.2 s prediction horizons..... | 66 |
| Table 3.5. Testing results of RBF SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.4 s prediction horizons..... | 67 |
| Table 3.6. Testing results of RBF SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.6 s prediction horizons..... | 68 |
| Table 3.7. Overall averages and standard deviations of the actual prediction horizons of SVM 1..... | 69 |

| | |
|--|----|
| Table 3.8. Testing results of second-order polynomial SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.2s prediction horizons. | 70 |
| Table 3.9. Testing results of the RBF SVM 1 and the RBF SVM2 using the 16 drowsy drivers in Testing Set B with lateral position and lateral velocity as input variables at 0.4s prediction horizons. | 73 |
| Table 3.10. Testing results of the RBF SVM 1 and the RBF SVM2 using the 16 drowsy drivers in Testing Set B with lateral position and lateral velocity as input variables at 0.6s prediction horizons. | 74 |
| Table 3.11. Overall averages and standard deviations of the actual prediction horizons of SVM 1 and SVM 2 tested against the 16 drowsy drivers in Testing Set B. | 75 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1.1. A two-dimensional example illustrating linearly separable data . | 16 |
| Figure 1.2. Margin for linearly separable data | 17 |
| Figure 1.3. A two-dimensional example illustrating non-linearly separable data . | 20 |
| Figure 1.4. Mapping input data into a higher dimension feature space . | 23 |
| Figure 2.1. Two driving maneuvers performed by drowsy driver #2 that appear to be quite similar. | 30 |
| Figure 2.2. The VIRTTEX driving simulator | 33 |
| Figure 2.3. Distance between the vehicle's center of gravity and the center line. | 34 |
| Figure 2.4. Percent of driving in different lateral positions for control driver 1. | 36 |
| Figure 2.5. Percent of driving in different lateral positions for control driver 5. | 37 |
| Figure 2.6. Percent of driving in different lateral positions for drowsy driver 3. | 38 |
| Figure 2.7. Percent of driving in different lateral positions for drowsy driver 14. | 38 |
| Figure 2.8. One hour of driving for drowsy driver 14. | 40 |
| Figure 2.9 Steering angle and lateral velocity signals during a lane deviation for drowsy driver 14. | 40 |
| Figure 2.10 Lateral acceleration and lateral velocity for drowsy driver 14. | 41 |
| Figure 2.11. Marked and unmarked lane deviations for control drivers. | 43 |
| Figure 2.12. Marked and unmarked lane deviations for drowsy drivers. | 44 |
| Figure 2.13. Duration of unmarked lane deviations for the 16 drowsy drivers. | 52 |

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

Most vehicle crashes are caused by drivers' fatigue or cognitive distraction [1]. In-car electronic devices such as computers, navigation systems, and mobile phones can lead to driver's inattention that causes drifting off road. A research by the New England Journal of Medicine [2] found that drivers who talked on a cell phone were four times more likely to be in an accident than drivers who didn't. Heated dispute with passengers, fiddling with a radio or climate control system, thinking about other things rather than focusing in driving can also lead to driver distraction. Official report of traffic accidents indicated that hazardous driving behavior, such as drunk and drowsy driving, was responsible for a high proportion of car accidents [3]. Drinking alcohol prior to driving greatly increases the risk of car accidents that leads to death. A driver who consumes a large amount of alcohol are more likely to be involved in an accident. Alcohol slows down the functions of the central nervous system and delay the function of a normal brain. This means that a person is unable to function normally. A person's information-processing skills (cognitive skills) and hand-eye coordination (psychomotor skills) are also affected by alcohol. The severity of alcohol impairment depends on the amount of alcohol present in the blood, which is based on a person's blood alcohol content percentage. According to National Highway Traffic and Safety Administration (NHTSA) report, drivers are considered to be alcohol-impaired when their blood alcohol concentration is 0.08 grams per deciliter (g/dL) or higher. In 2008, there were 11,773 fatalities in crashes involving a driver with a blood alcohol concentration of 0.08 or

higher (in the United States) [4]. Driver drowsiness is considered as a major contributing factor of road crashes. The statistics of fatalities and injuries due to traffic accidents become more serious year by year. According to NCSDR/NHTSA Expert Panel on Driver Fatigue and Sleepiness (1998), driver's drowsiness was responsible for 56,000 crashes each year (in the United States) resulting in more than 40,000 injuries and 1,550 fatalities per year.

As the performance of computers and vision systems improved, the road safety became one of the largest areas of research in the automobile industry. Many driver support systems were developed to prevent vehicle crashes or mitigate their effects. In early years, seat belt, air bags, and rumble strips (area of grooved pavement placed on the road shoulder) had a large contribution on road safety. Recently, new automobiles are equipped with sophisticated driver assistance systems. Advanced driver assistance systems (ADAS's) assist drivers for a driving process by monitoring the driving behavior and the vehicle surroundings to predict the dangerous situation in advance and take the desired action. These systems can be categorized according to its function to driver informing systems, driver warning systems, control-intervention systems and fully automatic control systems [5]. Moreover, DAS can be classified to longitudinal (such as rear sensing for parking, adaptive cruise control, forward collision warning, pedestrian detection and avoidance, and forward crash mitigation and avoidance—active braking) and lateral assistance systems (i.e. lane departure warning system, lane keeping assist systems, parallel parking assist, and blind spot monitoring and lane change).

Among these promising systems, lane departure warning systems (LDWS) have a tremendous potential to save lives. Lane departure warning system is an electronic system installed in a vehicle to warn the driver when the vehicle is about to veer from the lane and no turn signal is used (unintended lane leaving) [6]. The main goal of lane departure warning systems is to provide a suitable warning to drivers, without any external intervention, at an early stage so drivers can take the correct action to avoid run-off-road crashes.

During the past several years, great efforts were made to design and develop robust and reliable LDWS. There are three main functions of any lane departure prediction system, data collection, data processing, and the decision making. According to the literature, LDS's can be divided into three main classes based on the methodology applied; Algorithms based on calculating the time that a vehicle takes to cross the lane (Time to Lane Crossing TLC), on using vision sensors to estimate the position of the vehicle with respect to lane marking from the road scene, or on estimating the vehicle position using the driving variables (i.e. lateral position of the vehicle, steering angle, yaw rate, vehicle speed, vehicle lateral speed, etc.). Despite the method used, LDS should work properly in different road geometries (straight or curved), in different weather conditions and in any time of the day (day or night).

In some cases, the lane departure is a normal event that can be resolved by a simple steering correction. However, the lane departure can lead to a very dangerous situation. In 2003, 42,643 people were killed in the United States; the number of fatalities due to the vehicle's lane leaving was 25,000 lives [7] which represents more than 58.5%

of the total fatalities . In 2010, according to the National Highway Traffic Safety Administration [8], 32,885 died in motor vehicle traffic crashes in the United States, 17,389 –almost 53% died in the crashes as a result of abnormal lane departure on the road.

Different driving patterns can be seen daily in roadways. We can agree that people drive differently; some people drive in the center of the lane while others weave between lane lines. For some, this may indicate a distraction due to drowsiness, drunkenness, or using some electronics while driving, while for others this is a driving style. However, the challenge is how we can predict in advance unintended lane departures caused by distracted drivers and in the same time keep the number of nuisance alarms as small as possible.

Different techniques, as can be seen in the literature, have been used to develop lane departure warning systems that can assist drivers and reduce the number of fatal crashes. Among these, the methods that use machine learning techniques to learn driving variables to detect the driving patterns that lead to unintended lane departure. These techniques do not subject to mathematical approximations and image processing.

The most recent and powerful machine learning technique that is used for pattern recognition is the Support Vector Machine (SVM). SVM is a new supervised learning technique that based on statistical learning theory. SVMs are performing better, in many cases, than artificial neural network (ANN) [9]. SVM uses a quadratic programming which means the local minimum is always global minimum, unlike ANN which can be trapped by local minima.

All the above stated points plus the huge amount of data collected in Ford motor company give a big motivation and challenge to do research and contribute to this field of ongoing study to save lives.

1.2 Literature Review

Three main different techniques were used for developing lane departure systems. The main goal of this section is to review the literature related to the LDWS implemented in previous works and highlight the shortcomings associated to each method. The literature review is organized based on the technological method used for implementing the model.

1.2.1 Time to Lane Crossing-Based Techniques

Time to lane crossing-based technique was first proposed by [10]. TLC is defined as: the time that the vehicle takes to drift out of the road boundary assuming the current steering wheel angle is held constant and there is no further steering intervention by the driver. Assuming that the steering angle is constant, [11] proposed an algorithm to find the intersection between the vehicle path and the road lines and to calculate the time required by a vehicle to cross this point (TLC) using the predicted down range road geometry and the future vehicle path. The interpolation method was used to improve the results and reduce the error. According to the author, the interpolation scheme reduced the TLC error by 40%. In this paper, some factors that affect the down range road geometry finding and the future path of the vehicle prediction were also studied to assess their effect on the calculation accuracy of TLC.

Publication [12] developed a LDWS by using a lateral position and a speed of the vehicle to calculate (TLC). A vision based lane markers detection; lateral position estimation and a warning system were the main parts of the system. A lateral position of the vehicle was estimated by processing digitized images from a color video camera mounted on a side of a vehicle. Even though the system was acceptable and could detect lane markers in 60Hz, some human factors need to be added to improve the system. In the same way, [13] used a lane images contracted from a video camera and vehicle data to come up with data fusion algorithm that can robustly predict TLC. The vehicle position in the lane and the markers were constructed from a vision system, and sensors were used to collect the kinematic data of the vehicle. [14] Proposed a new method to reduce the false alarms resulted from a vision-based TLC-based lane departure prediction system. The new method utilized models of the driver behavior of directional sequence of piecewise lateral slopes (DSPLS) to discriminate between lane departure event and driver correction maneuver. The evaluation showed that this method could distinguish between the two events with an error rate of 17%.

Paper [15] conducted three experiments to compare the accuracy of TLC calculated in different ways. Three methods of computing TLC were examined. The trigonometric method which computed distance to line crossing (DLC) taking into account a curved path of the vehicle and divides it by the vehicle speed, TLC approximation using the first derivative of lateral position, and TLC approximation using the first and the second derivative of lateral position. In the first experiment, the TLC computed by the trigonometric method compared with the approximated TLC's in a

normal driving maneuver with no lane departure. The results showed that the second approximation which used the first and the second derivative of lateral distance was comparable with the trigonometric TLC, while the first approximation that used the first derivative of lateral position showed poor results. This means that the first approximation can't be used in studying a driver behavior. The other two experiments were conducted during a lane change maneuver and an intentional lane departure due to impaired driving. According to the author, the simple approximation gave acceptable results over a short time before a lane leaving occurs.

Many researchers studied the issue of accurate TLC computation. However, the absence of the vehicle state variables and the vehicle trajectory made it difficult to estimate an accurate value of TLC. Paper [16] examined different methods to estimate TLC based on the computation of distance to line crossing (DLC). A dynamic model with some approximations was used to calculate DLC and TLC in straight and circular roads in both zero and constant steering angle taking into account the steering characteristic of the vehicle. To solve the problem of road information unavailability, a state observer was proposed to estimate the road curvature assuming that the road curvature is almost constant. Moreover, by eliminating the slip motion of the vehicle, a kinematic model could be used to predict future vehicle location. The real data showed that to get acceptable results, vehicle dynamics must be considered.

In general, we can say that TLC is a simple powerful method to predict an unintended lane leaving. However, the major shortcoming of the systems that use this technique is the large number of false alarms especially with drivers tends to drive close

to the lane markers. These nuisance alarms result from the use of assumptions and approximations to computing TLC. Practically, it is not possible to estimate the exact value of TLC due to the non-linearity and the absence of road geometry and vehicle state information.

1.2.2 Estimating Vehicle Position with Respect to Lane Markings Using Vision

Sensors

Detecting the lane boundary from the road images is another technique for lane departure systems. Paper [17] implemented a two model vision-based algorithm for LDWS. The estimated model used perception-Net to estimate the vehicle's pose and the lane geometry while the warning model alarms when a lane crossing detected. By combining the classical vision-based lane detection approach with flow-based lane recognition technique, which measures the horizontal movement of road parallel structures, through a Kalman filter [18] could improve the performance of the system especially in a bad weather conditions. Publication [19] proposed a vision-based LDWS that utilized the edge information to express an edge distribution function. The edge distribution function was used to estimate the orientation of edge pixels and its local maximum to detect the lane boundary. Not far from that, [20] presented a brightness-adaptive image processing-based LDWS. To enhance the visibility of the lane lines in the images, a gradient approach was used and an appropriate threshold selected to split between the lane lines and the roadway.

Paper [21] developed a method to detect lane markers in a different road conditions. The method based on extracting the lane markers from the road image then

grouping them to detect the lane markers using the least square approach. [22] designed a real time DSP-based LDS that can work in a different weather conditions. A high performance digital signal processor was used to process digital signals of road images to detect lane marking and estimate the distance between the vehicle and the lane lines. These images captured by a video camera mounted on the windshield of a vehicle. To reduce the noise from the signals, median value and edge enhancing filters were used. According to the author, the system showed promising results and can be practically used.

A novel lane detection algorithm to detect marked roads in images was proposed by [23]. This algorithm combined the road geometry features determined by a geometrical model and the lane model matching done by Gabor filter (linear filter used for edge detection) to improve the accuracy and facilitate the computation process of a lane detection.

Most of the previous studies in LDWS's were dedicated to warn the drivers in highways. One of the few studies that focused on urban roads was demonstrated by [24]. In this work, Hough transform and B-Snake were used to detect the lane boundary from an edge image contracted by Canny edge detector. The system provided a warning signal based on the ratio between the left and the right displacement of the vehicle. To get rid of the errors resulting from the lane recognition process, [25] experimented a new method depended on detecting lane boundary points from forward looking images captured by a video camera mounted on the top left corner of the windshield. Using a vehicle model, the expected vehicle trajectories related to the detected boundary points and the

corresponding steering angel could be computed. Moreover, the time required for the vehicle to cross the lane could be estimated based on the expected trajectory and the speed of the vehicle. When this time exceeded a certain threshold, a lane departure tendency was predicted.

Publication [26] conducted a research to detect the lane departure using the road geometry model. The lane boundary was detected by processing the images acquired by a CMOS camera. To improve the real time processing, the settings of the dynamic region of interest were used. In a related work, [27] proposed a LDWS by applying Hough transform to detect the lines in a segmented regions in the lower part of a lane image (area of interest). A lane departure decision was made based on the distance between the lane lines. In the same way , by extracting and segmenting the region of interest from images captured by a video camera [28] designed a new LDWS. A lane boundary model was implemented using the Hough transformation and a subtractive clustering algorithm, a Kalman filter was used to predict the future position of the lane markers. This system issued a warning based on the position of the vehicle with respect to the lane boundary which determined from the camera parameters and the width of the vehicle.

Realistically, detection of lane markings from the road images is the most important task of this kind of LDWSs. Even though, this technique showed robustness in some cases of lined roadways, it is not applicable in unmarked roads. Moreover, beside the difficulty of real time detection of the lane lines many other obstacles may cause the use of this technique quite limited. These complications include: shadow caused by

objects outside the road like trees, lines masking by dirt, water, or other vehicles in the street, beside the challenge of distinguishing between the varieties of lane markings.

1.2.3 Estimating Vehicle Position Using Vehicle Variables

The third technique that was used for LDSs is the one based on estimating vehicle position with respect to the lane lines using driving variables (i.e. steering angle, yaw activity, speed of the vehicle, breaking, etc.). In an early work, [29] presented a Future Offset Distance warning system that uses a new alarm decision model. This model allows the vehicle to pass the lane boundary by applying an adaptive virtual lane boundary to reduce the number of false alarms. An alarm triggers when the lateral position of a vehicle is greater than the virtual lane boundary. [30] combined different methods to implement a LDWS. The lane departure event was detected by measuring the lateral displacement of the vehicle using the lane detection algorithm. Furthermore, to make the system more effective, a virtual lane boundary was proposed to warn the driver in a suitable time before the lane leaving actually occurs. Besides, to reduce the number of false alarms, warnings were suppressed if the driver set the turning signal or breaks. Risack used the TLC to detect the case when a vehicle cutting the curved lane to avoid the unnecessary warning.

Paper [31] employed images captured by a CCD camera to estimate the vehicle parameters. Image processing was used to obtain lane marking, to calculate the lateral offset and the lateral velocity of the vehicle by analyzing the images in a frequency of 30 frames/sec. A radial basis probability network (RBPN), related to fuzzy neural network FNN, was applied to distinguish between a normal lane changing and an unintended lane

departure then trigger a warning based on the level of dangerous expected. 5 seconds of a lateral velocity and a lateral lane position before each normal lane change and unintended lane leaving were devoted. The neural network trained with seven scenarios of video recording and five simulation scenarios. According to the author, after several learning and training, neural network was able to determine unintended lane departure by rate of 96.34%.

To differentiate between the intended and unintended lane departure, [32] designed a model using driving activities. The yaw rate and the lateral speed were used to estimate the driver's activity and trigger a warning based on the state of the driver.

1.3 Research Objectives and My Technical Contributions

The main goal of the dissertation is to explore an innovative approach to predicting unintended lane departure with minimum false alarms using some of the vehicle variables. My technical contributions include:

- I explored utilizing the nonlinear binary support vector machine (SVM) technique and the time series of vehicle variables to predict unintentional lane departure, which is innovative as no machine learning technique has previously been attempted for this purpose in the literature. I explored several sets of vehicle variables as inputs to the SVM. I experimented the linear, polynomial and RBF kernels for the SVMs, which are the most popular kernels in the literature. The SVM's predictive ability was experimentally optimized by finding the best SVM

kernel and its parameter values and the most appropriate set of vehicle variables, which turned out to be the lateral position and lateral velocity.

- The preliminary results showed that the SVM was able to predict most of the lane departures. However, a significant number of falsely predicted lane departures were observed. To minimize the number of falsely predicted lane departures, I developed a two-stage SVM training scheme: in which the first-stage testing results of a SVM were used in the second-stage of its training.

1.4 Introduction to Support Vector Machines

Support vector machine (SVM) is a supervised learning method based on statistical learning theory that can be used for data classification, regression and pattern recognition. SVMs can be applied to solve many complicated problems that cannot be solved by classical programming techniques due to the absence of a mathematical model (i.e. hand writing character recognition, speech recognition, data mining and knowledge discovery, image classification and several biomedical applications). The basic idea behind SVMs is to find the best hyperplane(s) that can separate n-dimensional data into a number (two in case of binary SVMs) of categories or classes. To make it clear, this section is dedicated to define some technical terms and explain the mathematical background of support vector machines.

1.4.1 Supervised and Unsupervised Learning

Machine learning is classified as supervised or unsupervised learning. Supervised learning is the type of learning in which training data are available and labeled with the

correct result (i.e. the input data and the corresponding output is known). The training data is in the form $\{(x_1, y_1)(x_2, y_2) \dots (x_l, y_l)\}$ in a dimension of $R^n \times R$ where x_i is the input data (known as samples, features, or attributes), y_i are the outputs or labels, and l is the number of examples. In other words, supervised learning aims to classify objects to one of a pre-specified set of categories or classes. Generally, supervised learning consists of two steps: training step, to learn classifier from training data, and testing step (known as generalization) to enable unseen objects to be identified as belonging to one of the classes. Examples of supervised learning techniques are decision trees, neural networks, and support vector machines. However, this is not always the case, and there is another type of learning in which the training data with pre-defined labels are not available. This is known as unsupervised learning or clustering. In this case, the program search for the similarity between samples of data in order to decide which objects should be grouped together without any prior information. This technique can be used in image segmentation, and speech coding. As a supervised learning technique, SVM is one of the powerful techniques that can be used in classification and pattern recognition. There are two classes of SVMs, linear and nonlinear. A linear SVM can only perfectly separate classes of data that are linearly separable by a hyperplane (in the case of binary SVM) or a set of hyperplanes (in the case of multi-class SVM), whereas a nonlinear SVM is capable of classifying data with more complex structure that are not linearly separable. Strictly speaking, no real-world classification problem is linearly separable. For predicting lane departure events, a binary nonlinear SVM was needed because there were

only two class labels: 1 (lane departure) and -1 (not lane departure). In the next few subsections, we briefly review how a binary linear and nonlinear SVMs work.

1.4.2 Linear Support Vector Machines

This section introduces a brief dissection of the mathematical background of linear support vector machines. Only two-class (binary) SVMs are discussed for both separable and non-separable data.

1.4.2.1 Separable classes

Let \mathbf{X} and \mathbf{Y} be the input set and output set of the data to be classified, respectively. For binary classification, output is $y_i \in Y = \{1, -1\}$. Let's suppose there are l training examples with n features, (\mathbf{x}_i, y_i) where $\mathbf{x}_i \in \mathbf{X} \in R^n, i = 1, 2, \dots, l$. A data point to be classified is denoted $\mathbf{x} \in \mathbf{X}$. The main idea of training a SVM is to find the best line in two-dimensional cases, the best plane in three-dimensional cases, or the best hyperplane in higher than three-dimension that separates the two classes of data with a maximum margin. Such a line, plane or hyperplane is called the decision function. Figure 1.1 illustrates an example of the maximum margin separation between two classes of two-dimensional data where the label (1) represent the points related to class C_1 (blue dots), while (-1) for those of C_2 (red dots) [33]. The equation for the line separating the data into two classes is

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (1.1)$$

and the decision function can be written as

$$D(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \quad (1.2)$$

where \mathbf{w} is a 2×1 the weight vector perpendicular to the line (determine the direction of the separating line) and b is the bias (determines the exact position of the separating line). The data are linearly separable because a line can divide the two data classes without any misclassification. As shown in the Fig. 1.1, this hyperplane is not unique. However, only one of them achieves maximum separation (solid line) and should be chosen because it leaves more space in both sides, so that data can move a bit without a risk of being wrong classified. This is called a maximum margin classifier, and it gives a less chance of misclassifying data when it works with unknown data (testing data); this concept is known as the generalization performance of the classifier.

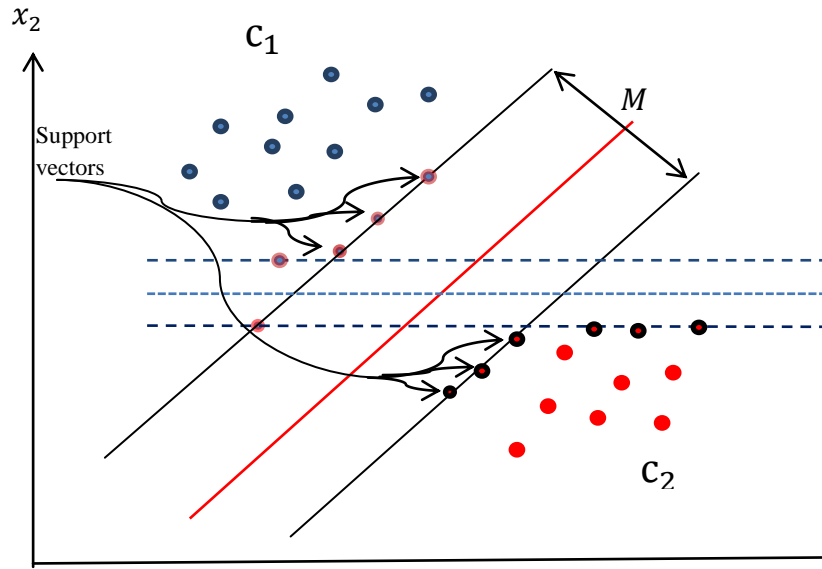


Figure 1.1. A two-dimensional example illustrating linearly separable data [34].

We can scale \mathbf{w} , b so that the value of $D(x)$ at the nearest points to the margin (known as support vectors) is equal to 1 at C_1 , and -1 at C_2 .

$$\mathbf{w} \cdot \mathbf{x} + b = 1 \quad (1.3)$$

$$\mathbf{w} \cdot \mathbf{x} + b = -1 \quad (1.4)$$

and, $\mathbf{w} \cdot \mathbf{x} + b \geq 1 \quad \forall \mathbf{x} \in C_1$

$$\mathbf{w} \cdot \mathbf{x} + b \leq -1 \quad \forall \mathbf{x} \in C_2$$

Thus: $\mathbf{w}(\mathbf{x}_1 - \mathbf{x}_2) = 2$

for two support vectors $\mathbf{x}_1, \mathbf{x}_2$ on each side of the hyperplane, the margin is given by projecting the vector $(\mathbf{x}_1 - \mathbf{x}_2)$ onto the normal vector to the hyperplane i.e. $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ (Figure

1.2) from which we get

$$M(\text{margin}) = \frac{2}{\|\mathbf{w}\|} \quad (1.5)$$

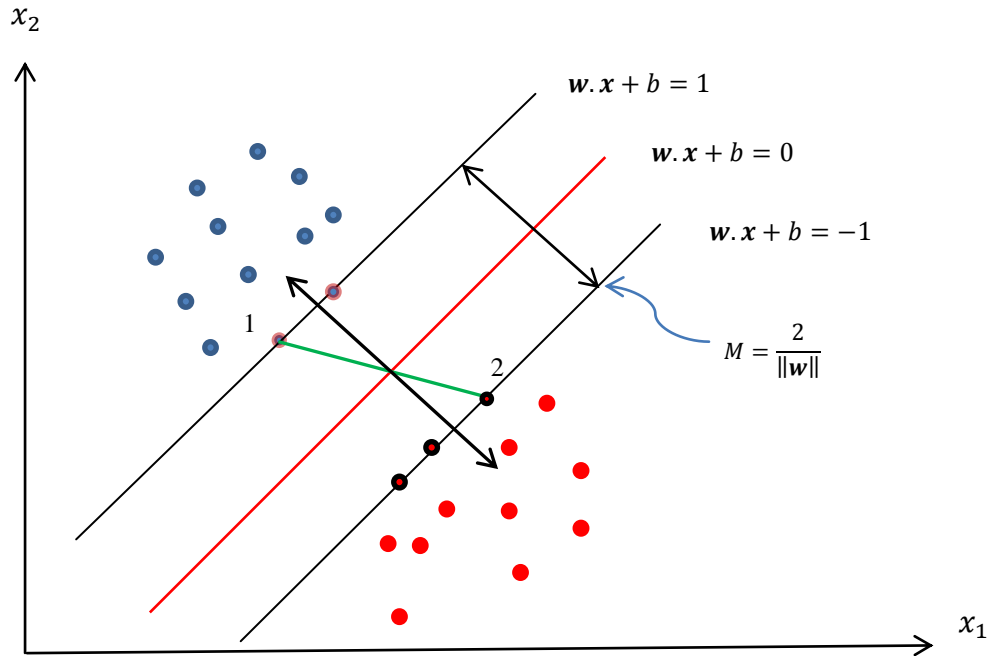


Figure 1.2. Margin for linearly separable data [34].

The task now is to compute the parameters \mathbf{w} and b for maximum margin. Maximizing the margin is equivalent to the minimization of the function [35]:

$$j(\mathbf{w}) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) \quad (1.6)$$

Subject to the constraints:

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 \quad (1.7)$$

This is a quadratic optimization problem subject to linear constraints. The objective function to optimized is

$$L(\mathbf{w}, b) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^l \alpha_i [y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1] \quad (1.8)$$

Where L is the Lagrange function and α_i is the Lagrange multipliers. To find the optimum solution, this requires the derivative of L with respect to \mathbf{w} and b vanishes,

$$\frac{\partial L}{\partial b} = 0 \text{ gives;}$$

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (1.9)$$

$$, \frac{\partial L}{\partial \mathbf{w}} = 0 \text{ gives;}$$

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad (1.10)$$

by substituting (1.9), (1.10) into (1.8) we get

$$w(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{j,i=1}^l \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad (1.11)$$

Subject to the constraints

$$\alpha_i \geq 0, \text{ and}$$

$$\sum_{i=1}^l \alpha_i y_i \geq 0$$

The decision function is

$$D(\mathbf{x}) = \text{sign}[\sum_{j=1}^l \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}) + b] \quad (1.12)$$

For every training point there is a Lagrange multiplier α_i that can be either zero or positive. The points in which the Lagrange multiplier $\alpha_i > 0$ are called *support vectors*

and lie on either of the hyperplanes $\mathbf{w} \cdot \mathbf{x} + b = \pm 1$. The eq. (1.6) is a strict convex, this means the optimal hyper plane classifier of a support vector machine is unique [35]. One remark is, the importance of SVMs comes from the theoretical bounds on the generalization error which has two features: the upper bound on the generalization error does not depend explicitly on the dimensionality of the input space, and the bound is minimizing by maximizing the margin M .

1.4.2.2 Non-Separable Classes (Soft Margin Classifier)

Consider the case where the training data are not separable as shown in Figure 1.3. In this case, there is no possibility to draw a hyperplane that can completely separate the two classes; moreover, the above algorithm cannot be applied. Three classes of training data can be seen in this case: data are correctly classified, data are correctly classified (in the right side of the separating line) but falling inside the band enclosed by $\mathbf{w} \cdot \mathbf{x} + b = 1$, and $\mathbf{w} \cdot \mathbf{x} + b = -1$, and data that are misclassified. To solve this problem with a minimal number of errors, we allow some amount of slackness in constraints (1.7) when necessary by introducing new variables,

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 - \xi_i, \quad \xi_i = 1, \dots, l \quad (1.13)$$

$$\xi_i \geq 0, \quad \xi_i = 1, \dots, l \quad (1.14)$$

The variables ξ_i are known as slack variables. It is clear that if $0 \leq \xi_i \leq 1$ the data point lies between the margin and the correct side of hyperplane, and if $\xi_i \geq 1$, data point is misclassified (Fig. 1.3). The objective now is to maximize the margin, and at the same

time keeping the points were $\xi_i \geq 0$ as small as possible. Thus, in this case the objective function to be minimized is

$$j(\mathbf{w}, b, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^l \xi_i \quad (1.15)$$

subject to the constraints (1.13), (1.14). The parameter c in eq. (1.15) is a regularization constant, a positive constant that can be estimated using a cross-validation technique or can be found experimentally. As the value of c changes, the norm $\|\mathbf{w}\|$ changes correspondingly [36].

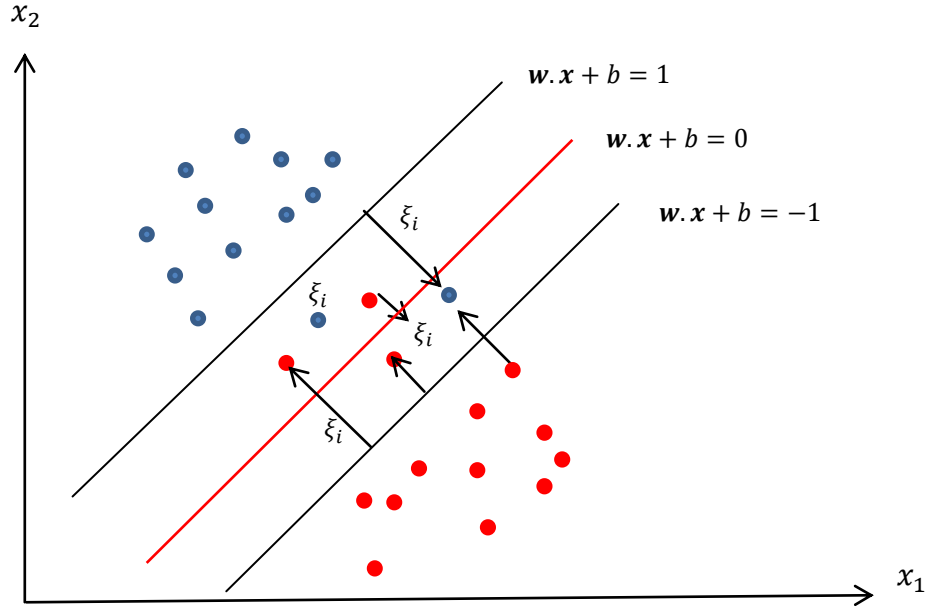


Figure 1.3. A two-dimensional example illustrating non-linearly separable data [35].

This is also a quadratic programming problem, and its lagrangian is given by

$$\begin{aligned} L(\mathbf{w}, b, \xi, \mu) = & \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^l \xi_i - \sum_{i=1}^l \mu_i \xi_i \\ & - \sum_{i=1}^l \alpha_i [y_i(\mathbf{w} x_i + b) - 1 + \xi_i] \end{aligned} \quad (1.16)$$

Where μ_i are the Lagrange multipliers introduced to enforce positivity of the ξ_i . To find the optimum solution, this requires the derivative of L with respect to \mathbf{w} and b vanishes

$$\frac{\partial L}{\partial b} = 0 \text{ gives;}$$

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (1.17)$$

$$, \frac{\partial L}{\partial \mathbf{w}} = 0 \text{ gives;}$$

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad (1.18)$$

$$, \frac{\partial L}{\partial \xi_i} = 0 \text{ gives,}$$

$$c - \mu_i - \alpha_i = 0, \quad i = 1, \dots, l \quad (1.19)$$

$$\sum_{i=1}^l \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i], \quad i = 1, \dots, l \quad (1.20)$$

$$\mu_i \xi_i = 0, \quad i = 1, \dots, l \quad (1.21)$$

$$\mu_i \geq 0, \quad i = 1, \dots, l \quad (1.22)$$

$$\alpha_i \geq 0, \quad i = 1, \dots, l \quad (1.23)$$

Thus the task now is to maximize (1.16) subject to constraints (1.17), (1.18), (1.19), (1.20), (1.21), (1.22), and (1.23). Now, by substituting the above equality into the Lagrangian, we can see that, neither ξ_i nor their Lagrange multipliers appear in the dual problem

$$\text{Maximize } w(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{j,i=1}^l \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad (1.24)$$

subject to,

$$0 \leq \alpha_i \leq c, \quad i = 1, \dots, l \quad (1.25)$$

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (1.26)$$

Equation (1.19) combined with (1.21) shows that if $\alpha_i < c$ then $\xi_i = 0$. If $\xi_i > 0$ (when the data points fall inside the band) the corresponding Lagrange multipliers will have the upper bound of c .

1.4.3 Nonlinear Support Vector Machines

Linear support vector machines cannot handle many of complicated classification tasks due to computational power limitation. For binary class data that are only nonlinearly separable, a nonlinear binary SVM is required. It first maps, via a function ϕ , the l data points in the input space \mathbf{X} to a higher (can be infinite) dimensional space so that the mapped data in the new space \mathbf{H} (called the feature space) become linearly separable. Figure 1.4 shows how the input data is mapped to the feature space. The feature space is a vector space where the dot product is applicable. The binary SVM's decision function is

$$\mathbf{X} \rightarrow \mathbf{H}$$

$$\begin{aligned} \text{i. e.} \quad & \mathbf{x}_j \rightarrow \phi(\mathbf{x}_j) \\ & \mathbf{x}_j \cdot \mathbf{x} \rightarrow \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}) \end{aligned}$$

And the decision function (1.12) becomes

$$D(\mathbf{x}) = \text{sign}[\sum_{j=1}^l \alpha_j y_j (\phi(\mathbf{x}_j) \cdot \phi(\mathbf{x})) + b] \quad (1.27)$$

It is quite clear that the input data appear in the decision function (1.12) in the form of inner product $\mathbf{x}_j \cdot \mathbf{x}$ and in (1.27) as the inner product of $\phi(\mathbf{x}_j) \cdot \phi(\mathbf{x})$. Thus, rather than explicitly mapping the input data into a higher dimension a space and performing a linear SVM classification, we can operate in the input space using the so called Kernel function

$K(\mathbf{x}_j, \mathbf{x})$ which implicitly represents the inner product of the mapped data in the feature space.

$$K(\mathbf{x}_j, \mathbf{x}) = \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}) \quad (1.28)$$

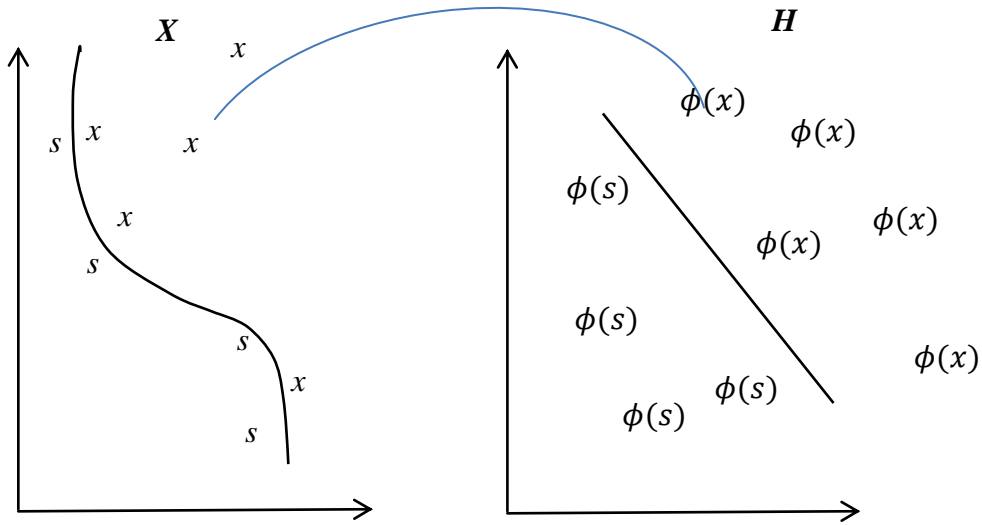


Figure 1.4. Mapping input data into a higher dimension feature space [36].

Mathematically, the valid Kernel must satisfy the Mercer's theorem:

for any $g(\mathbf{x})$, $\mathbf{x} \in \mathcal{C}$ such that,

$$\int_{\mathcal{C}} g^2(\mathbf{x}) d\mathbf{x} < \infty \quad (1.29)$$

there must be the case where,

$$\iint_{\mathcal{C}} K(\mathbf{x}, \mathbf{x}') g(\mathbf{x}) g(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \geq 0 \quad (1.30)$$

Thus, the Kernel should be a positive semi-definite. In non-linear SVMs we need first to choose a suitable kernel function, and then maximize the function

$$\text{Maximize } |\alpha \quad w(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{j,i=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (1.31)$$

$$\text{subject to,} \quad 0 \leq \alpha_i \leq c, \quad i = 1, \dots, l \quad (1.32)$$

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (1.33)$$

And substitute the optimum α_i in the decision function (1.27)

$$D(\mathbf{x}) = \text{sign} \left[\sum_{j=1}^l \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}) + b \right]$$

1.4.4 Kernel Functions

There exist many kernel functions in the literature. The commonly used ones include the linear kernel, polynomial kernel, exponential kernel, and radial basis function (RBF) kernel, to name a few [37].

- *Linear Kernel*: is the simplest Kernel function that given by the inner product as following

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' + k \quad (1.34)$$

where the constant, k is optional.

- *Polynomial Kernel*: is a non-stationary Kernel and data should be normalized when using this Kernel. A polynomial of degree, d can be written as:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + k)^d \quad (1.35)$$

where, k is a constant. It leads to the linear kernel when $d = 1$.

- *Gaussian (radial basis function) kernel*:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma^2}\right) \quad (1.36)$$

where σ is a design parameter.

- *Hyperbolic tangent (Sigmoid) Kernel*: the sigmoid function is the activation function which used in neural networks.

$$K(\mathbf{x}, \mathbf{x}') = \tanh(a\mathbf{x}^T \mathbf{x}' + k) \quad (1.37)$$

where, a is the slop, and k is a constant.

- *Exponential Kernel*: is similar to a radial basis Kernel.

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{\sigma^2}\right) \quad (1.38)$$

Choosing the most appropriate kernel function is an important step for any application. Nevertheless, there presently lacks a general method to find it. The kernel function selection is application-dependent and hence must be sought with a certain degree of trial-and-error effort. Performance of the SVM also depends highly on its design parameters, including the parameter of the kernel.

1.4.5 Kernel Selection

Choosing the kernel function is a key issue when using a learning technique. The kernel function $K(\mathbf{x}_j, \mathbf{x})$ implicitly represents the inner product of the mapped data \mathbf{x}_j, \mathbf{x} in the feature space $\{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_N(\mathbf{x})\}$ where N is the dimensionality of the Reproducing Kernel Hilbert Space [38]. Therefore, the choice of the a good kernel means the choice of the effective representation of the data. The problem of choosing the kernel

for the SVM, or the appropriate data representations for learning, is an important one. There is no a general method to find a good data representation; however, [39] has shown that for a given data representation there is a systematic method for kernel estimation using semi-definite programming. In the literature we can find many Kernel functions that can be used with non-linear SVMs.

1.4.6 Parameter Estimation

Parameter estimation is an important step in SVMs. SVMs algorithms usually depend on some parameters, the regularization parameter c which controls the tradeoff between margin maximization and error minimization and the kernel parameters which appear in the non-linear mapping into feature space such as σ in RBF and exponential kernel and d in the polynomial kernel. A cross-validation technique is usually used to assess SVMs classifiers with given kernel arguments and regularization constants. Moreover, SVMs parameters and kernel parameters can be chosen experimentally.

1.4.7 Data Preprocessing

Data preprocessing is an important step in machine learning systems and can affect the results. Raw data usually tend to be noisy, inconsistent or impossible data combination due to the loosely control in data collection. Different data preprocessing technique can be applied to correct the data. Data pre-processing includes data cleaning, normalization, transformation, feature extraction and selection [40]. Data cleaning, for example, can be used to remove noise and correct inconsistency in data. Data normalization can be applied, where data are scaled to fall within a specified range (i.e. 1-

to 1 or 0 to 1). This can improve the accuracy and efficiency of algorithms. Moreover, in data collection, many features may be used to represent the data, however only some of them can lead to good results. The process of selecting the appropriate variables and removing the irrelevant and redundant variables is called feature selection.

Data reduction technique is helpful in analyzing reduced data set. Complex data analysis on huge amounts of data may take a very long time, making such process impractical and time consuming.

1.4.8 Features Selection

The first step in machine learning systems is to select the appropriate set of training data that characterize the task to be learned. This step is known as features selection, variables selection, or attributes selection. Features selection has an enormous impact on the success of learning algorithms to recognize complex patterns and make intelligent decisions based on data. It is very important to select the most effective features rather than all available features.

CHAPTER 2 A TWO-STAGE SVM TRAINING SCHEME FOR LANE DEPARTURE PREDICTION

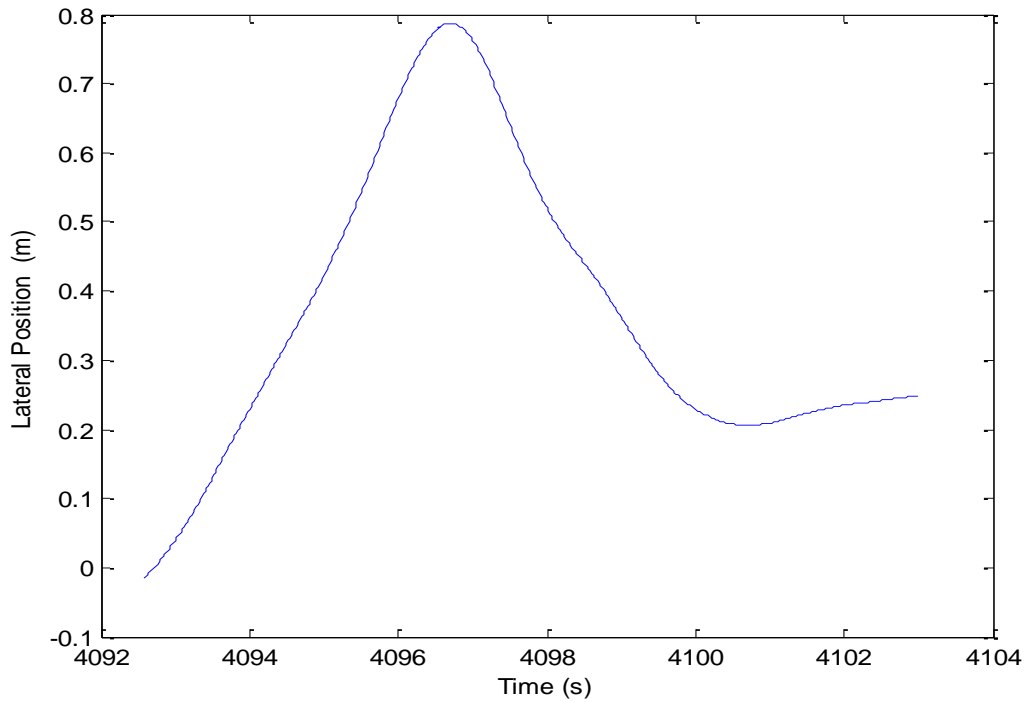
The main goal of this dissertation is to develop a robust and reliable unintentional lane departure prediction algorithm that can work in different conditions with minimum false alarms using some driving variables based on support vector machines (SVMs). In this chapter we show how the two-stage support vector machines (SVMs) training scheme can provide enhanced unintentional lane departure prediction.

2.1 The Two-Stage SVM Training Scheme

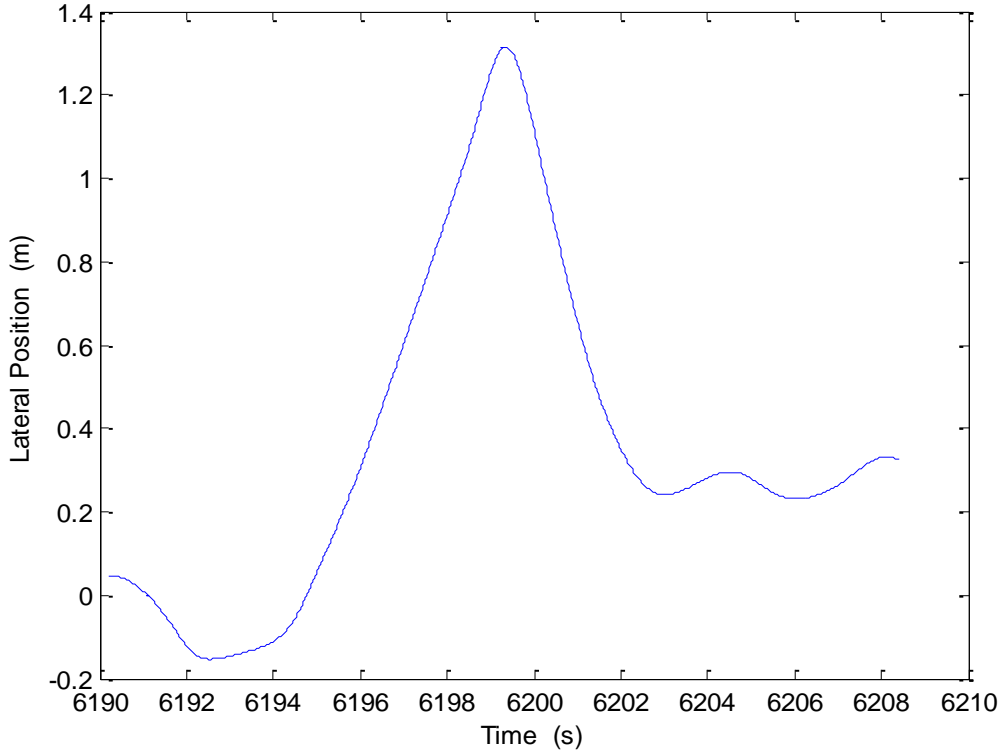
In a two-class classification application, which is the case of this study, a SVM is trained with input-output examples to find the best decision function to separate the two-class test data with minimum error. This section describes the steps that we applied to implement a two-stage SVM training scheme for lane departure prediction.

Initially, we explored a set of variables as inputs of a SVM to develop a lane departure prediction system. The preliminary results showed that the SVM was able to predict the majority of the lane departure events. However, there were a quite significant number of falsely predicted lane departures. We carefully analyzed the results and found that most of the SVM errors were made when the vehicle was close to the inner edges of either side of the lane boundaries. Because these situations resemble real lane departure patterns, they caused the SVM to misclassify. To better understand this type of situations, let us see two specific cases. Figure 2.1 shows two different types of driving maneuvers in terms of vehicle lateral position that were recorded during drowsy driver #2's

experiment. The width of the simulated lane was 3.81 m. The displacement of the vehicle from the center of the lane is called a lateral position of the vehicle. A lateral position of 0 m means the vehicle is in the center of the lane while a positive or negative lateral position means the vehicle is deviated toward the right or left side of the lane, respectively. A simulated 2000 Volvo S80 was used in our study and its body width is 2.19 m. Hence, the vehicle is out of the lane when the lateral position is greater than 0.81 m or less than -0.81 m. Fig. 2.1(a) displays a driving pattern when the driver was trying to prevent the vehicle to cross the lane boundary (i.e., maintaining the lateral position to be less than 0.81 m) even though the vehicle was very close to the boundary. Clearly, this is not a lane departure event. Fig. 2.1(b) shows a similar driving pattern that is a lane departure event. The high similarity in cases like these posed a challenge to the SVM.



(a) Driving maneuver pattern 1 (not a lane deviation event)



(b) Driving maneuver pattern 2 (a lane deviation event)

Figure 2.1. Two driving maneuvers performed by drowsy driver #2 that appear to be quite similar.

The cases like those in Fig. 2.1 motivated us to develop a two-stage SVM training scheme: in which the first-stage testing results of a SVM are used in the second-stage of its training. More specifically, for the first-stage training, the driving data of all the drowsy and control drivers that we had (see Section 2.2.1 below) were divided into three sets, namely Training Set A, Testing Set A, and Testing Set B. Training Set A contained training examples constructed using only (randomly) selected lane departure event data. Some of these examples were labeled as lane departure while the rest were marked as non-lane departure (see Section 2.3.1 below for detail). Because of the way this training

set is formed, the number of data points in it as percentage of the total number of data points is very small. Next, each of the driver's driving record was divided at the middle point of his/her driving time and the first half time record became Testing Set A while the second half became Testing Set B. Training Set A was used to train a SVM and Testing Sets A and B were employed to test the trained SVM, leading to an initial testing result (note that this result is the final testing result if the SVM is supposed to be trained in one single stage). This completed the first stage of the training. There are only four possible classification outcomes: (1) true negatives (correctly predicted non-lane departures), (2) false negatives (lane departures failed to be predicted), (3) true positives (correctly predicted lane departures), and (4) false positives (falsely predicted lane departures). While the initial testing result could be very good, it most likely would contain classification errors partially because the SVM failed to differentiate the similar driving patterns shown in Fig. 2.1. Subsequently, we carried out the second-stage training of the SVM. The training data for this stage, named a Training Set B, were entire Training Set A plus a number of randomly selected false positives in the initial testing result that were related to Testing Set A only (i.e., they had nothing to do with Testing Set B). After the second-stage training, the SVM was tested by Testing Set B, which yielded the final testing result of the SVM.

Due to the additional stage of training, the resulting SVM is expected to outperform a similar SVM undergoing only one stage of training. We will show the performance comparison results later. For convenience, the two-stage and one-stage trained SVMs are named SVM 2 and SVM 1, respectively.

2.2 Experiment Data

2.2.1 Data Source for SVM 1 and SVM2

The SVM was trained and tested using the driver experiment data generated by VIRTual Test Track Experiment (VIRTTEX), a hydraulically powered 6-degrees-of-freedom moving base driving simulator at Ford Motor Company. The experiments were conducted by a group of Ford researchers for evaluating four different human machine interfaces for lane departure warning around 2004 [41], which is about seven years before the work reported in this paper began. This simulator emulated a 2000 Volvo S80. The test participants, all were licensed drivers and signed the consent form, were divided into two groups - drowsy group and control group. There were 32 subjects in the former group who were deprived of sleep for almost one day before starting a three-hour simulated drive while the 6 control drivers having a full night of sleep drove the same simulated vehicle for 20 minutes.

The drive was on a simulated 514-km long, dry, four-lane (two for each direction) US interstate under nighttime conditions. The road was divided by a median or concrete barrier. The test participant was asked to drive in the right lane only. Traffic density was very low. Vehicle variables, including the common ones (e.g., speed, lateral position, and steering angle), were collected at a sampling period $T = 0.02$ s [42].

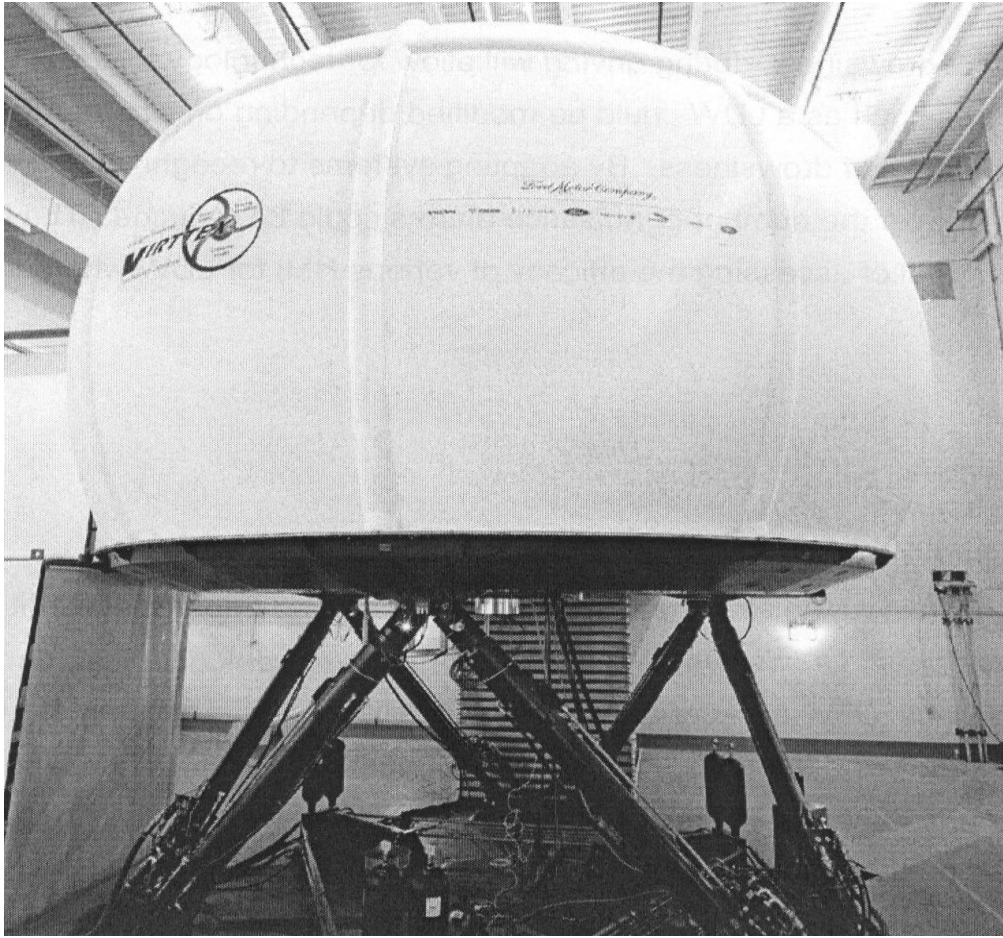


Figure 2.2. The VIRTTEX driving simulator [42]

We used the experiment data representing 16 drowsy drivers and 6 control drivers for the SVM lane departure prediction study. For each driver, the time series at the beginning of the experiment when vehicle speed was below 20 MPH were removed, so were the data at the end of the experiment after the brake was fully applied. The remaining time series were used in the training and testing of the SVMs. There were a total of 3,508 lane departures for the 16 drowsy drivers and only 23 for the 6 control drivers (two had none).

2.2.2 Description of Variables

This subsection gives definitions and formulas of the most common variables that we used in this lane departure prediction system.

- *Distance from the center line*: is the distance between the vehicle's center of gravity (c_g) and the center line (see Figure 2.3).

- *Lateral position*: is the lateral position of the vehicle in the lane. It can be calculated using the following equation:

$$\text{Lateral position} = \text{Distance from centerline} * \text{feet2meter} - \frac{\text{Lane width}}{2} \quad [m] \quad (2.1)$$

where $\text{feet2meter} = 0.3048$,

$$\text{Lane width} = 3.81 \quad [m]$$

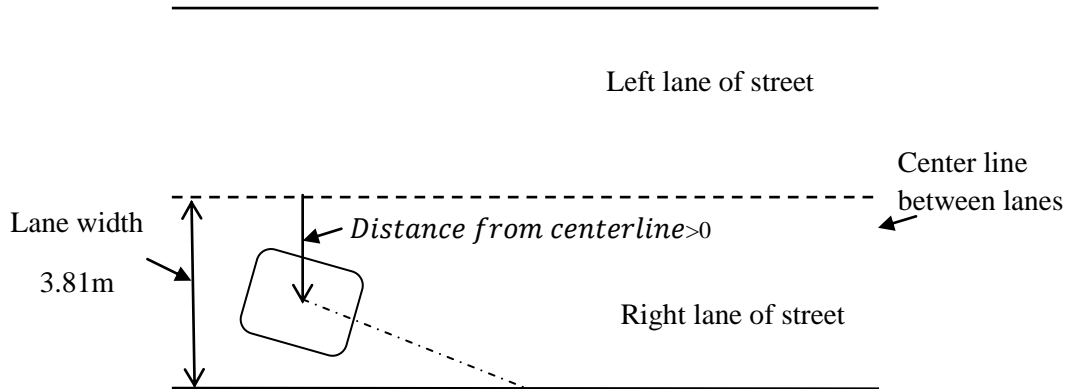


Figure 2.3. Distance between the vehicle's center of gravity and the center line.

-*Yaw angle*: is the vehicle's yaw angle in global coordinate system in degrees. To align to zero yaw for vehicle driving forward, we need to subtract 90° from *Yaw angle*.

- *Lane deviation*: indicates if a vehicle inside or outside a lane. This variable assigned value 0 if a vehicle still inside the lane and 1 when the vehicle is outside the lane. A vehicle is considered outside a lane when a lateral position is greater than 0.81 m or less than -0.81 m.

- *Lateral velocity*: is the lateral velocity of a vehicle. The lateral velocity of a vehicle is calculated using the following formula:

$$Lateral\ velocity(t) = \frac{Lateral\ position(t) - Lateral\ position(t-T)}{T} \quad (2.2)$$

where $T = 0.02$ is the sampling period.

2.2.3 Data Analysis

This section gives an overview about the recruited drivers and their style of driving based on the collected data. Moreover, Data needs to be analyzed to decide which variables that have a large contribution to lane departure so it can be used as inputs to the lane departure system [43]. There were many variables that were collected in VIRTEX lab at Ford Motor Company.

Figures, 2.4 and 2.5 show the percent of driving positions for the control drivers 1 and 5, respectively. The data shows that driver 1 had no any lane departure, while driver 5 drifted eight times out of the lane. It is clear that the control driver 1 drove most of the time in the middle of the lane (almost 80% within 0.2 m from the center of the lane), while the control driver 5 drove most of the time (96%) in the left side of the lane. Known that all drivers were ordered to drive in the right lane, control driver 5 hugged the

left side of the lane. This style of driving is more confusing because it shows that the driver is about to leave the lane while it is not. This can cause more false predictions of lane departure. Figures, 2.6 and 2.7 show the percent of driving positions for the drowsy drivers 3 and 14, respectively. The data shows that drowsy driver 3 departed the lane 58 times, while the drowsy driver 14 runs 389 times out of the lane boundary. Figure 2.7 indicates that drowsy driver 14 drove most of the time (more than 79%) in the left side of the lane and 31.5% of the time in more than 0.5 m lateral position. This explains the high number of lane departure the driver had committed. In case of drowsy drivers we cannot

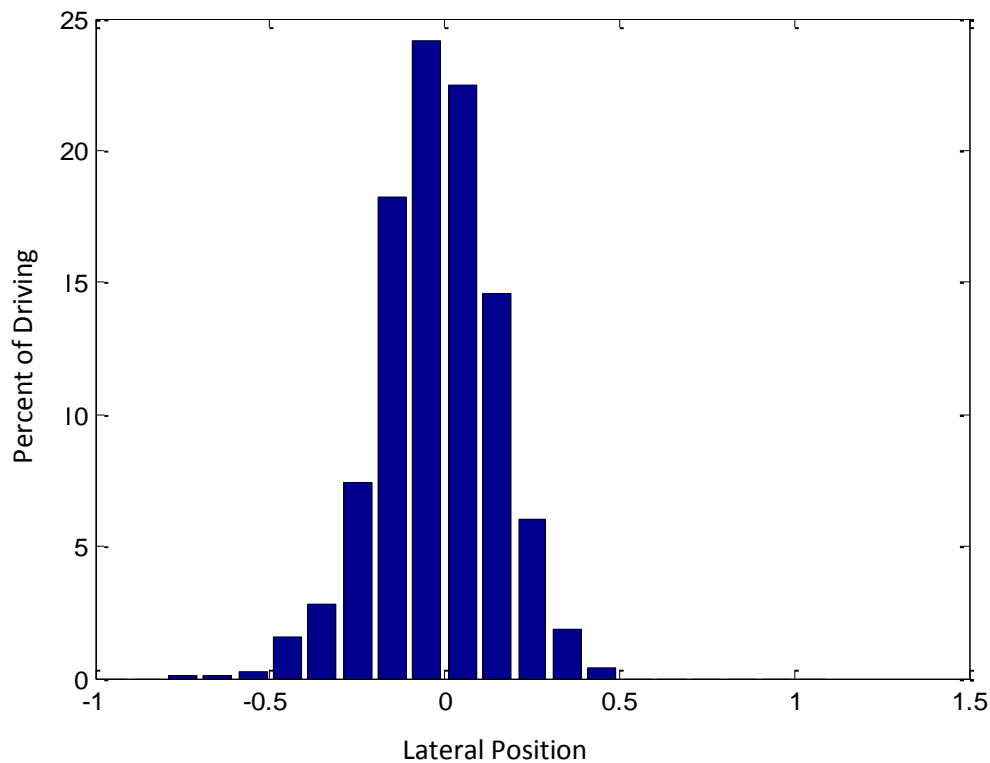


Figure 2.4. Percent of driving in different lateral positions for control driver 1.

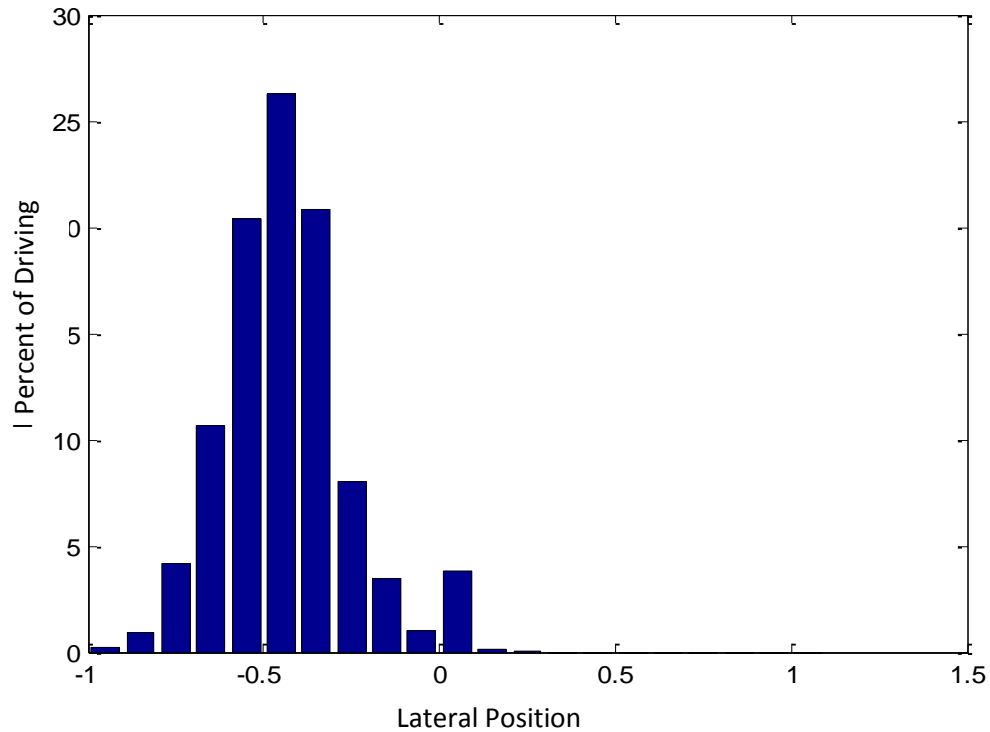


Figure 2.5. Percent of driving in different lateral positions for control driver 5.

attribute this to only the way of driving, the degree of fatigue due to drowsiness had a big influence on this way of driving. The conclusion that we could come up with was: a vehicle lateral position is one of the most important features correlated to a vehicle lane leaving.

More driving features were investigated to select the most appropriate variables for our system and remove the irrelevant and redundant variables. An example of a driver 14 shows different driving variables that can give a good notion about the better variables that can be used as input to the lane leaving prediction systems.

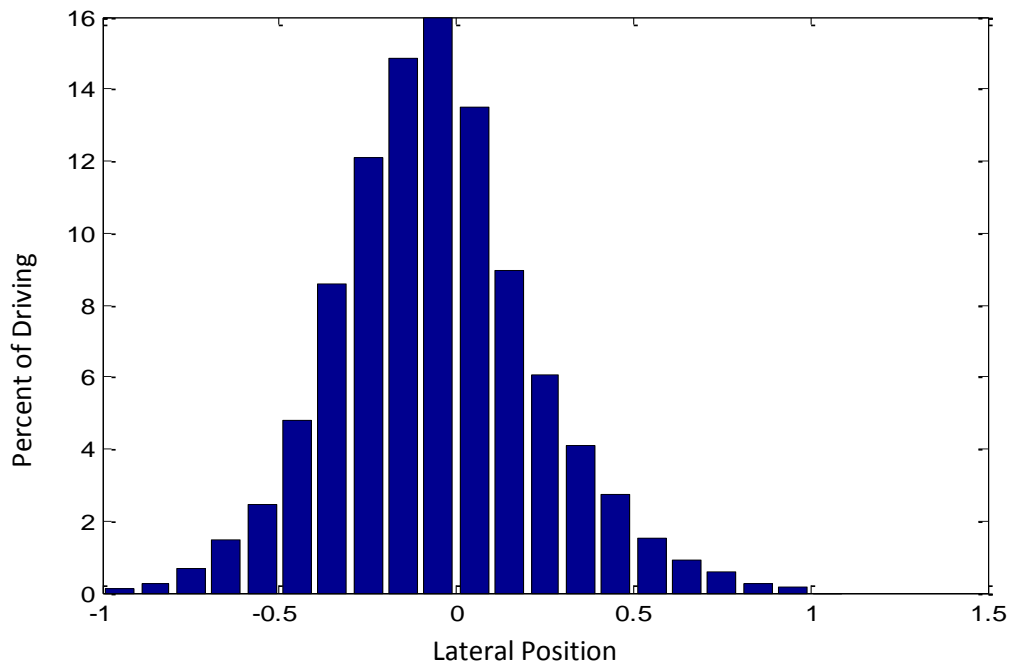


Figure 2.6. Percent of driving in different lateral positions for drowsy driver 3.

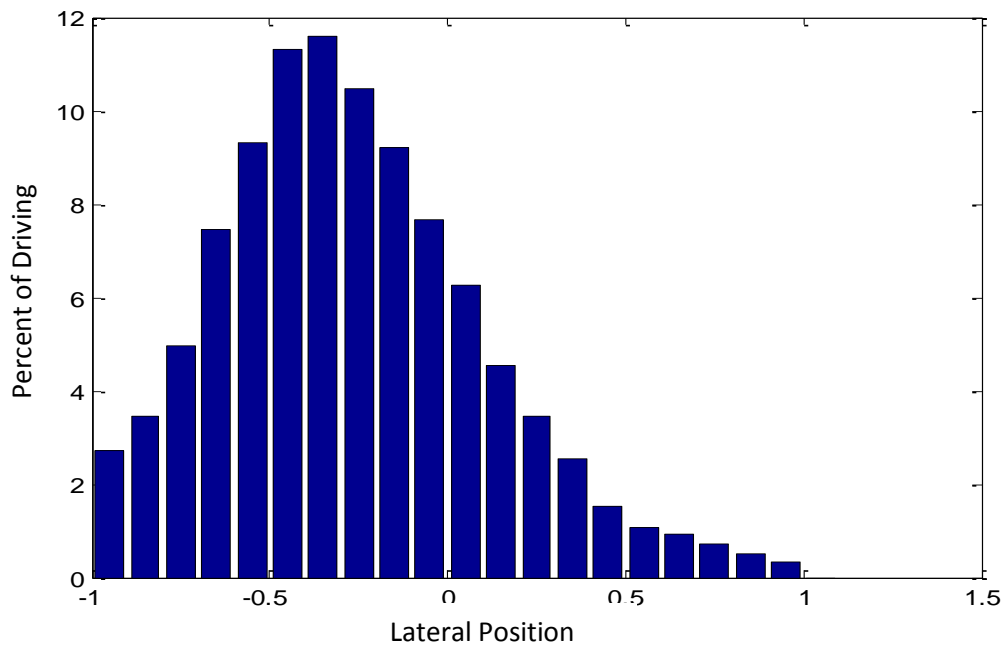


Figure 2.7. Percent of driving in different lateral positions for drowsy driver 14.

Example for drowsy subject 14

Figure 2.8 shows some features for the drowsy driver 14. As mentioned before, there is no fixed time for lane deviation. Long lane deviations show smooth transition in features while short lane deviations have abrupt features transition. Figure 2.9 focuses on steering signal and lateral velocity during a short lane deviation. The figure shows that during the lane leaving the driver exhibited approximately a sinusoidal steering pattern in order to fast return the vehicle back to the lane then straighten it. Furthermore, the lateral velocity during the vehicle return can be higher than that during the departure. Such patterns can only be learned (by machines) when a reasonable sized window of samples is used. Thus for all practical purposes when using the SVM for a lane departure, an entire window of samples is used as input instead of a single sample.

As many variables were collected, reasonably not all combinations of them may be used. As can be seen in Figure 2.10 the lateral velocity and the lateral acceleration are quite correlated. As lateral acceleration increases, the lateral velocity also increases and vice versa.

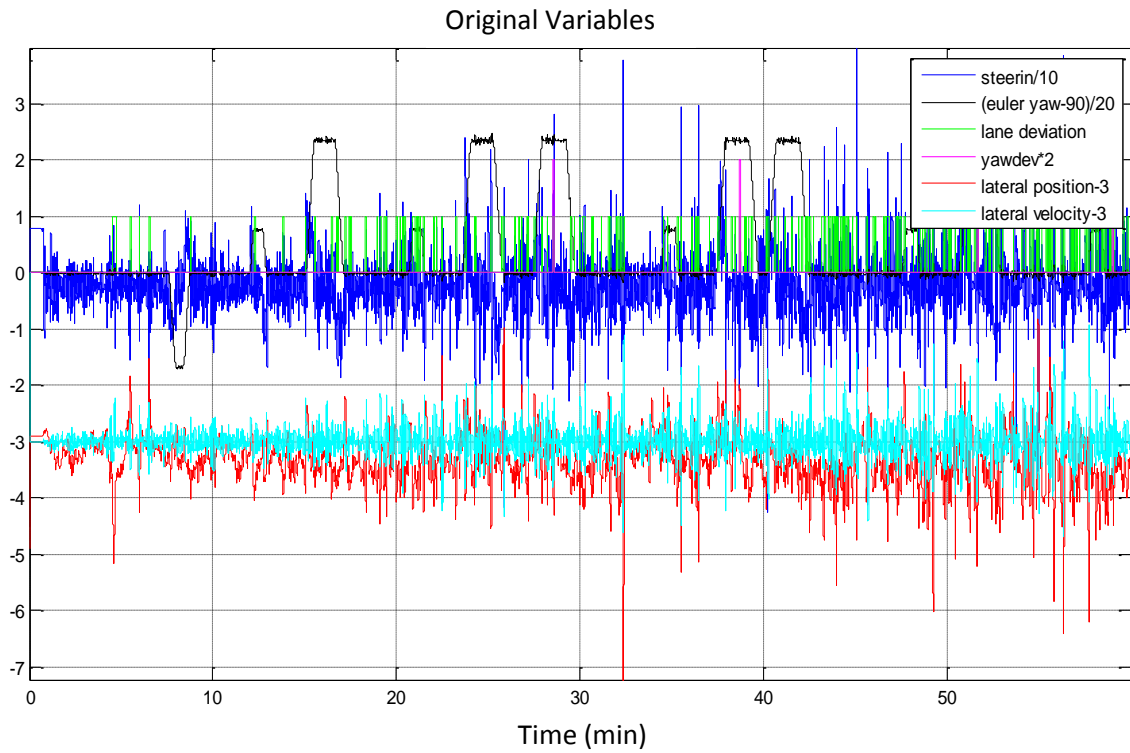


Figure 2.8. One hour of driving for drowsy driver 14.

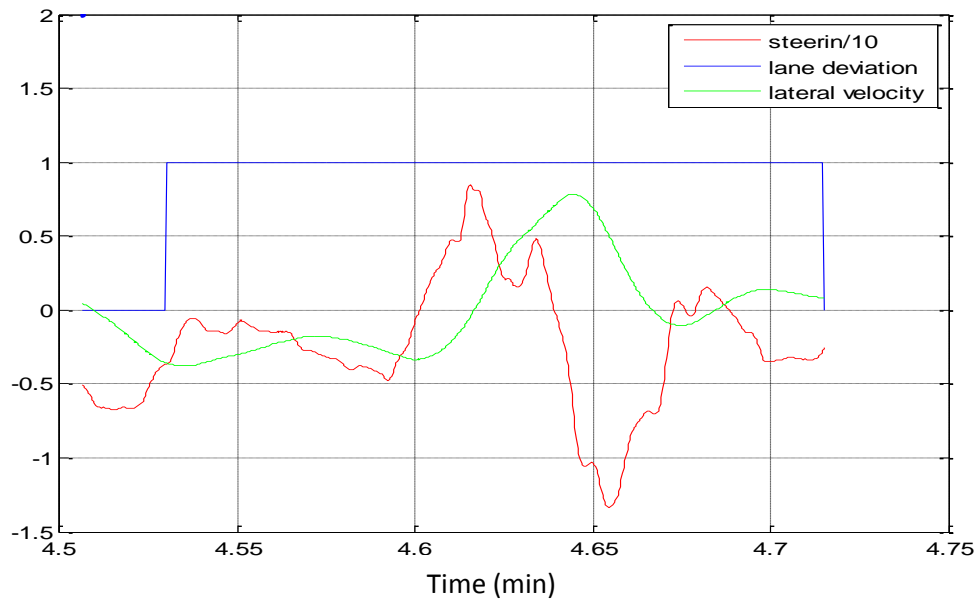


Figure 2.9 Steering angle and lateral velocity signals during a lane deviation for drowsy driver 14.

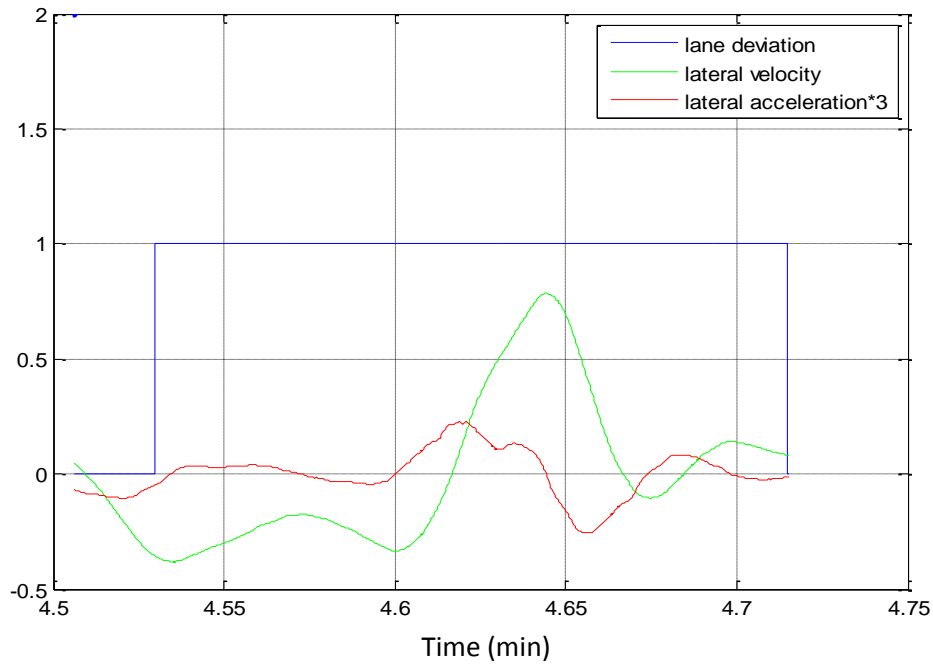


Figure 2.10 Lateral acceleration and lateral velocity for drowsy driver 14.

2.2.4 Data Cleaning

Data cleaning is an important step that can affect the results. Raw data usually tends to be noisy and inconsistent. The first step was to investigate the data of the six control drivers and sixteen drowsy drivers to see if the data were noisy or inconsistent. Two types of inconsistency were discovered; unmarked lane deviations and marked non-lane deviations. The unmarked lane deviation is a state in which the lateral position of the vehicle exceeds 0.81 m ($lateral\ position \geq 0.81$) and the vehicle is considered inside the lane ($lane\ deviation = 0$). The lateral-position threshold at which the vehicle is considered outside the lane is 0.81. The marked non-lane deviation is a state in which the

lateral position of the vehicle is less than 0.81 m (*lateral position* < 0.81) and the vehicle is considered outside the lane (*lane deviation* = 1). The inconsistency of the data can impact the results; therefore, deep investigations were done to clean the data. Figures, 2.11 and 2.12 show the numbers of marked and unmarked lane deviations for both control and drowsy drivers, respectively. Figure 2.13 indicates the repetition of different durations of unmarked lane deviations.

Two types of unmarked lane deviations can be seen in the data:

- The first unmarked lane deviations are the instances that preceding real lane deviations. Those are too short in time and appear more often in the data. We considered those as lateness in marking lane deviations, and the solution was to mark these samples as deviated samples.
- The second unmarked lane deviations are the ones that separated from real lane deviations. Those are too rare and last for a long time. The consistency in the lateral position indicated that, those unmarked lane deviations were real lane deviations, and the solution is to mark them as lane deviations.

The second type of data inconsistency is the marked non-lane deviation. There are three types of marked non-lane deviations:

- The first marked non-lane deviations are the instances that follow real lane deviations when the vehicle returning back to the lane. This type of wrong labeling has no effect on the results.

- The second marked non-lane deviations are the instances that preceding real lane deviations. Those are short in time and appear more often in the data. This type of error considers that the vehicle outside the lane before it truly exceeds the lane line and the solution is to mark these instances as non-lane deviations.
- The third marked non-lane deviations are the ones that separated from real lane deviations. Those are too rare and last for some seconds. To avoid the effect of these inconsistent data on the results, we excluded these parts of data.

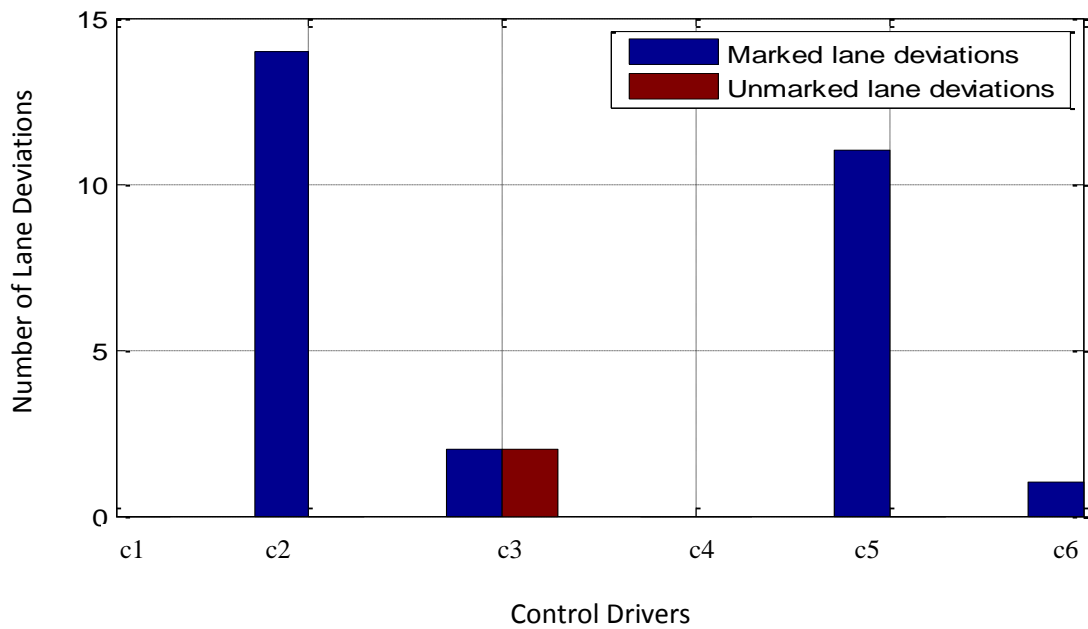


Figure 2.11. Marked and unmarked lane deviations for control drivers.

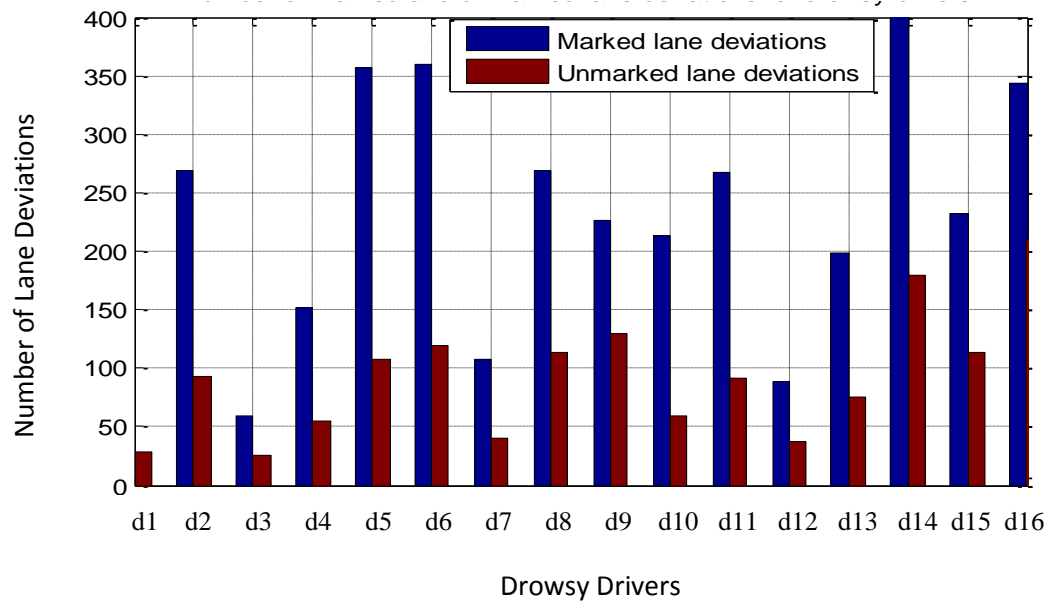
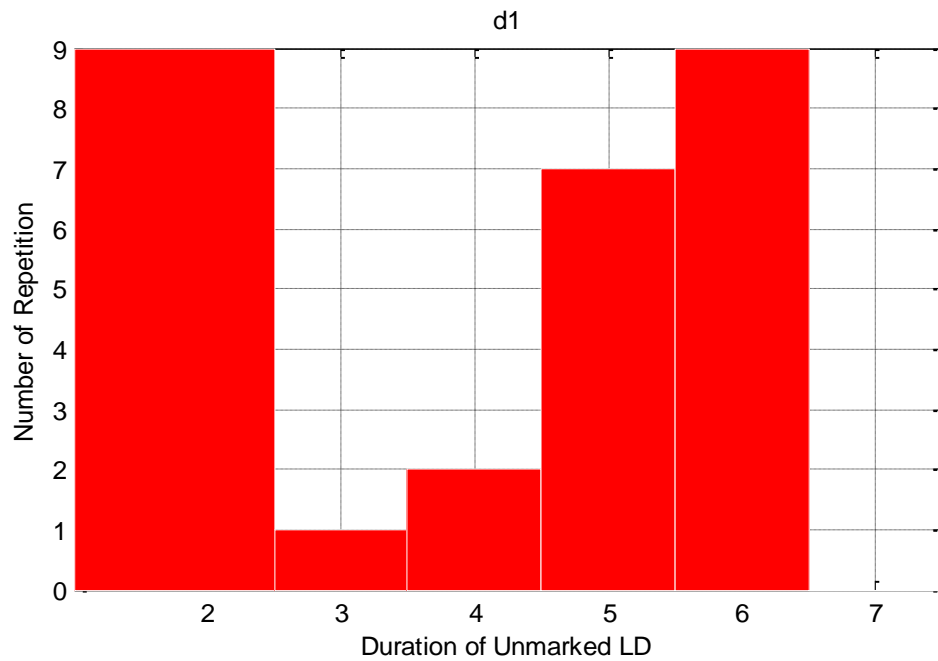
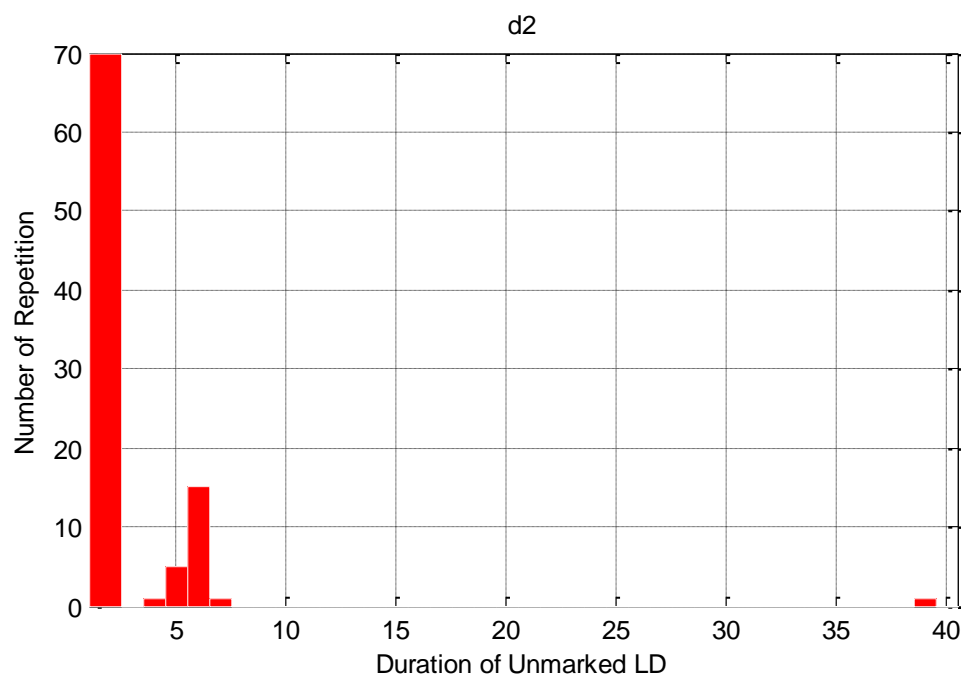


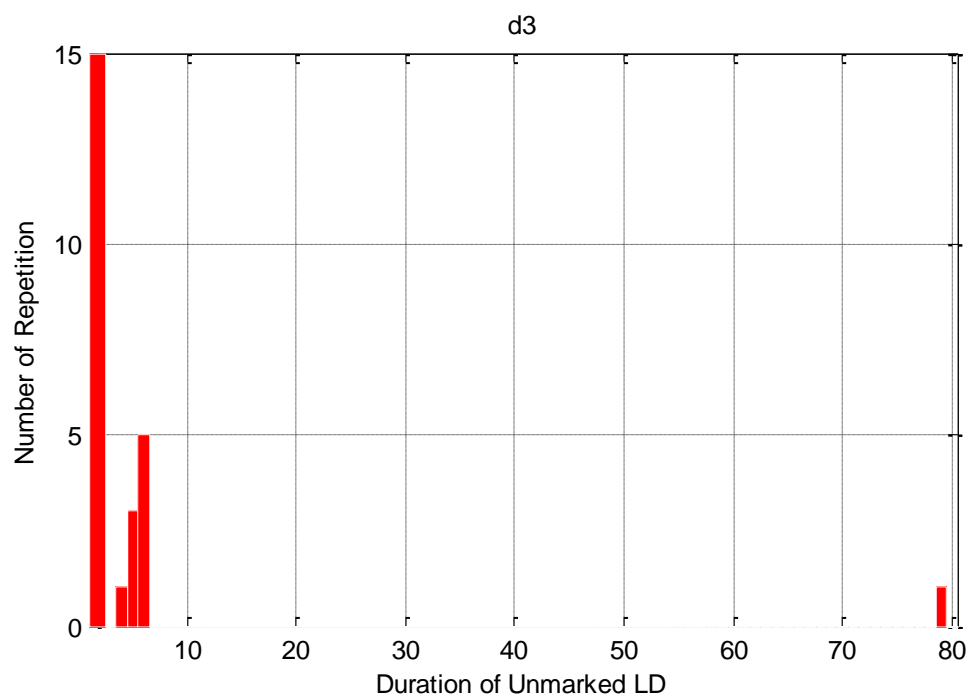
Figure 2.12. Marked and unmarked lane deviations for drowsy drivers.



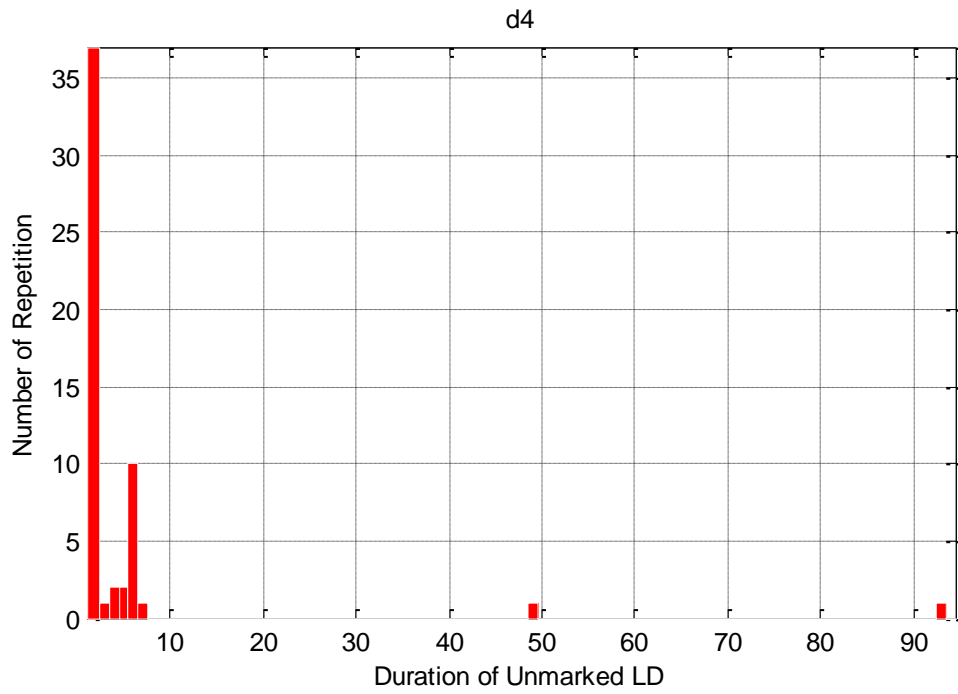
(a) Duration of unmarked lane deviations for drowsy driver 1



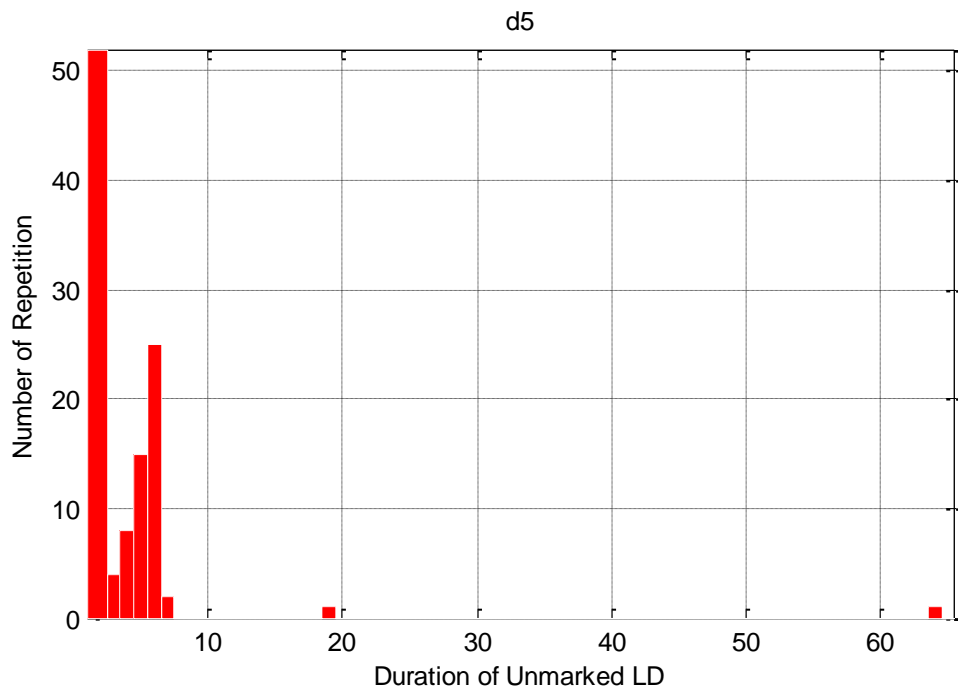
(b) Duration of unmarked lane deviations for drowsy driver 2



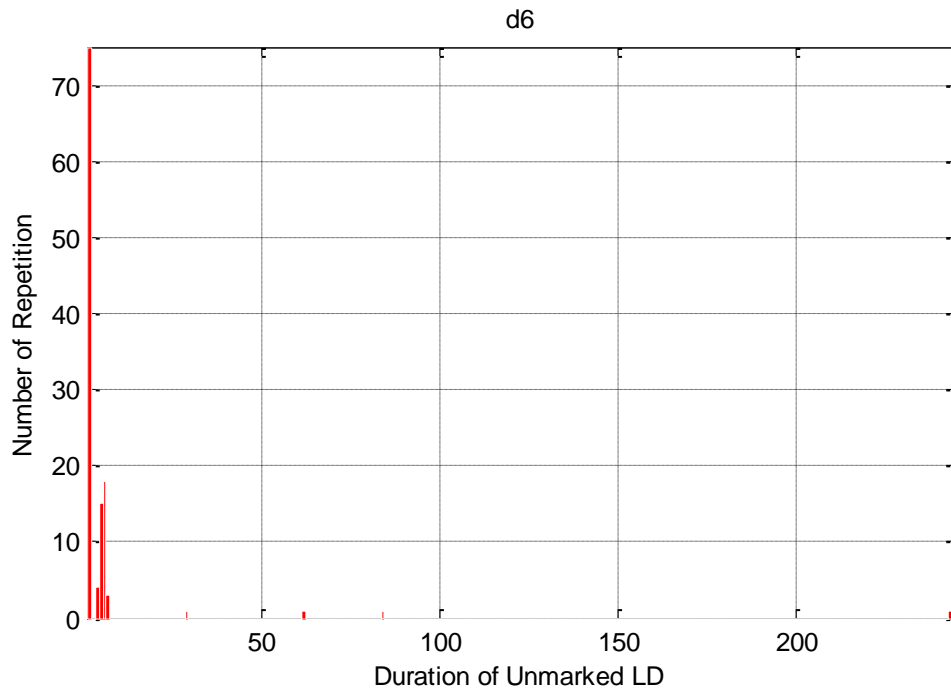
(c) Duration of unmarked lane deviations for drowsy driver 3



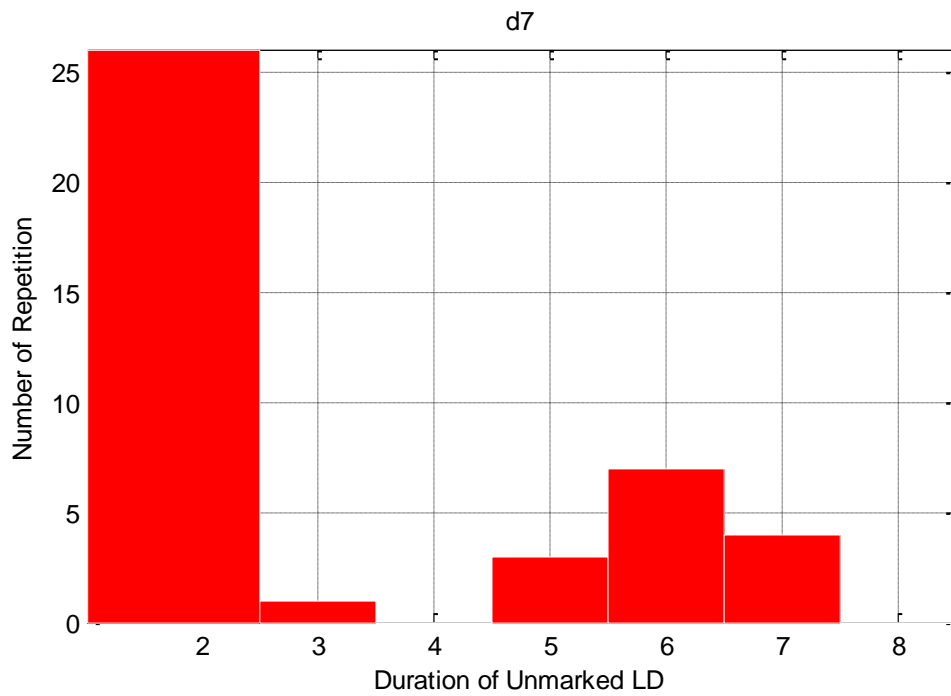
(d) Duration of unmarked lane deviations for drowsy driver 4



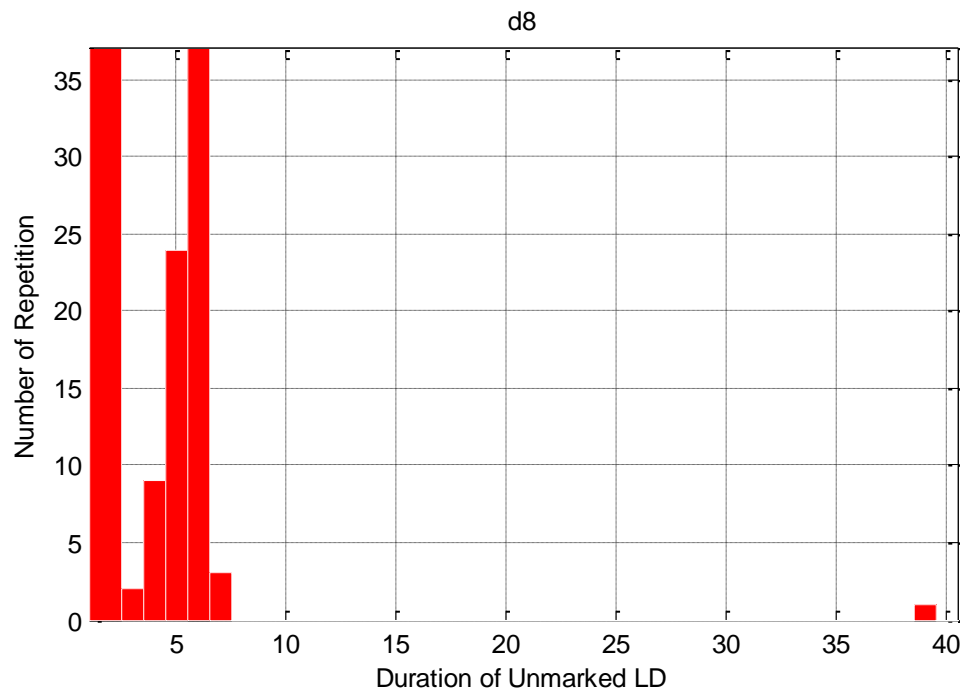
(e) Duration of unmarked lane deviations for drowsy driver 5



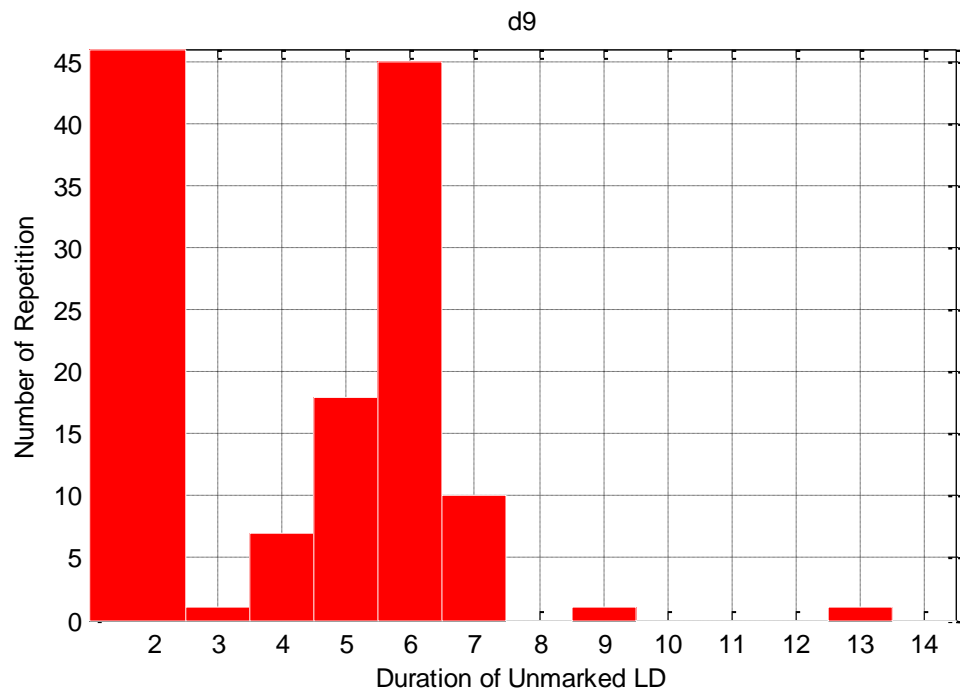
(f) Duration of unmarked lane deviations for drowsy driver 6



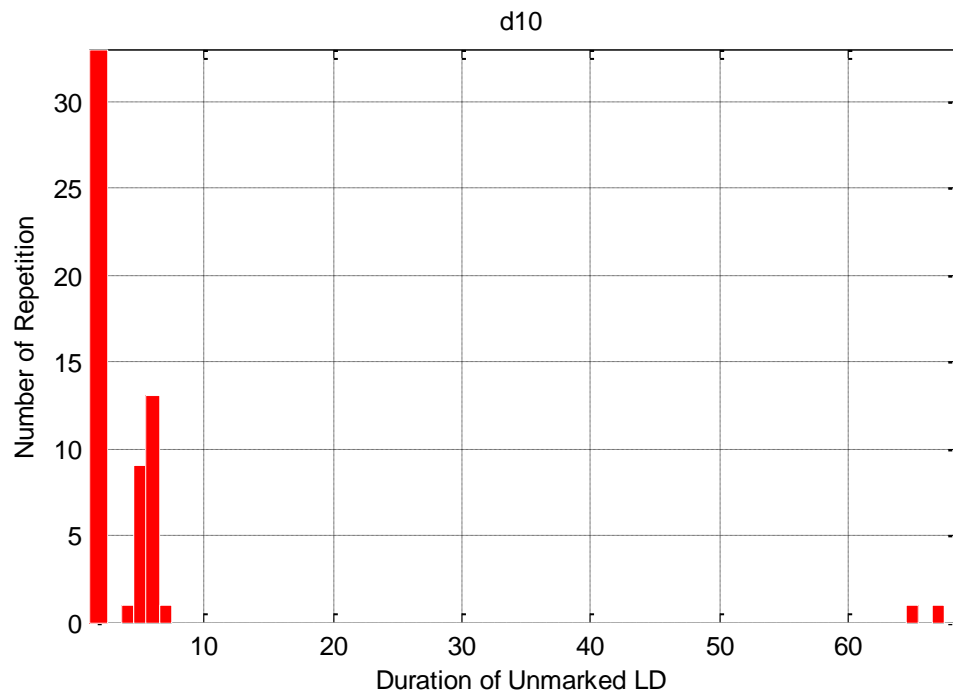
(g) Duration of unmarked lane deviations for drowsy driver 7



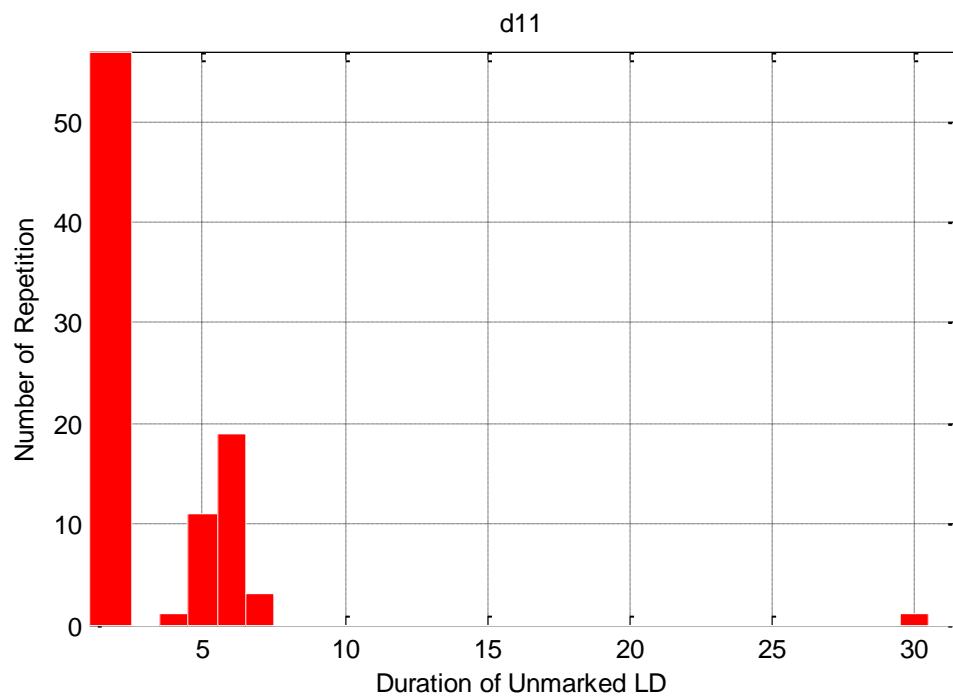
(h) Duration of unmarked lane deviations for drowsy driver 8



(i) Duration of unmarked lane deviations for drowsy driver 9



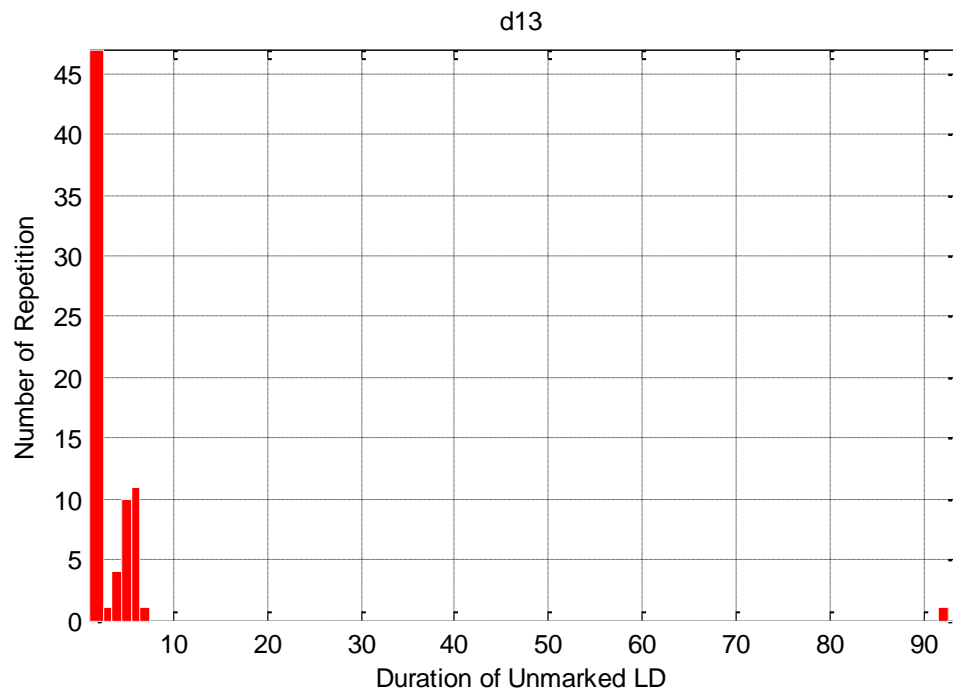
(j) Duration of unmarked lane deviations for drowsy driver 10



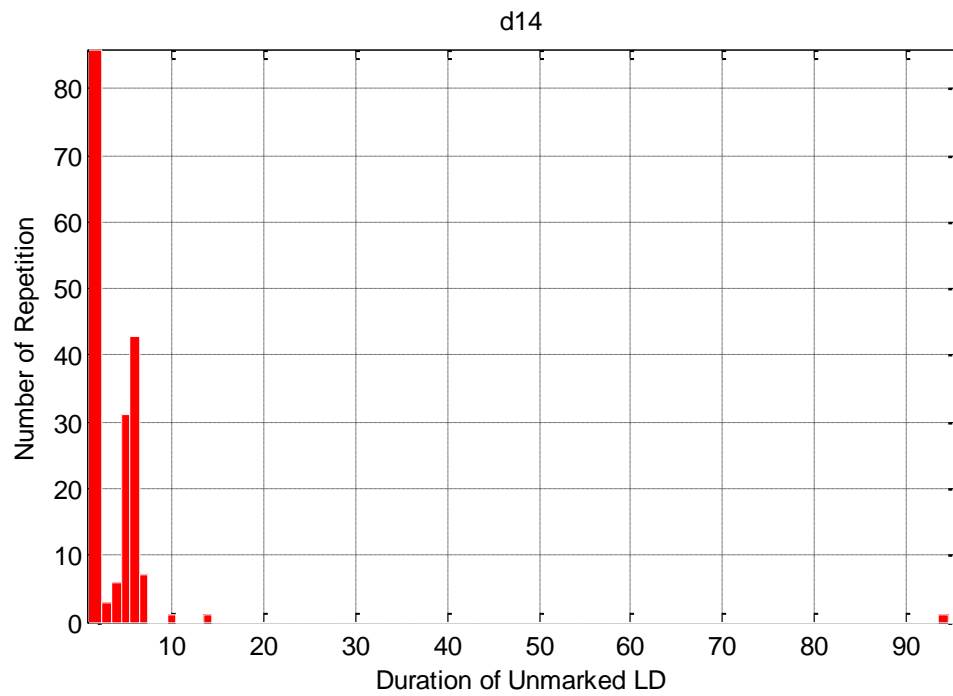
(k) Duration of unmarked lane deviations for drowsy driver 11



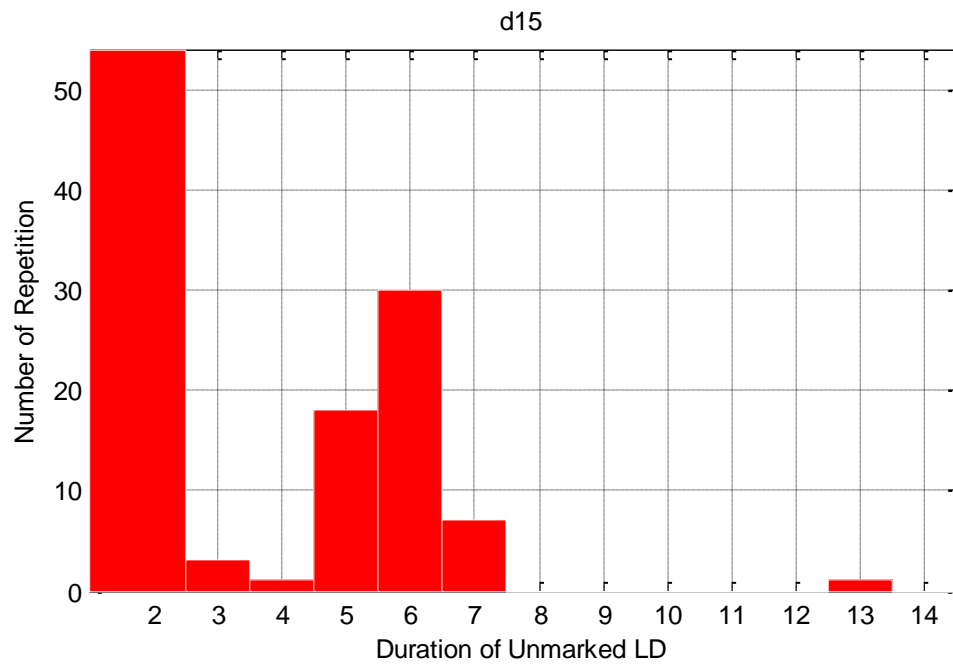
(l) Duration of unmarked lane deviations for drowsy driver 12



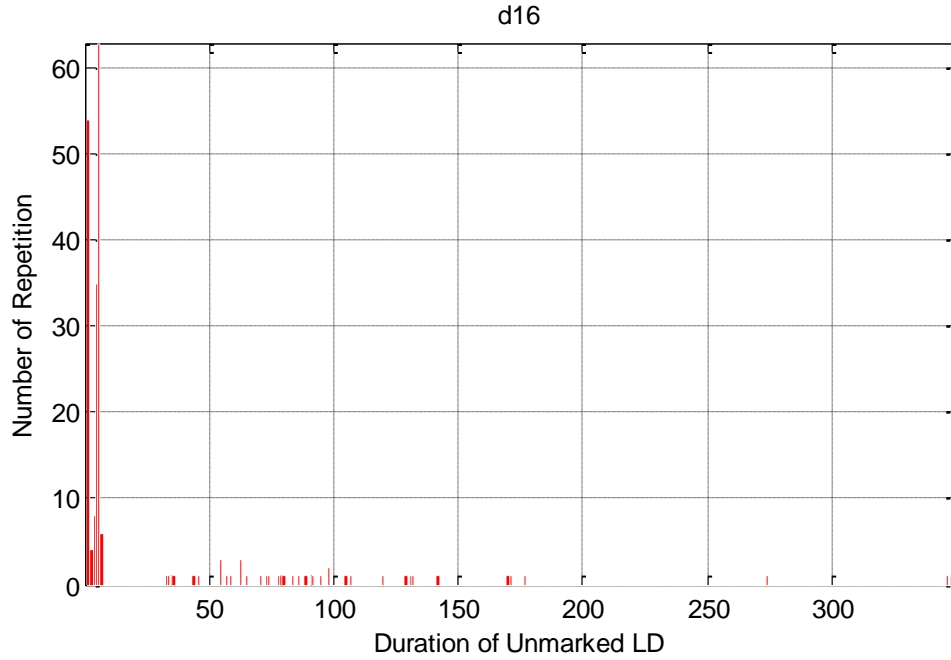
(m) Duration of unmarked lane deviations for drowsy driver 13



(n) Duration of unmarked lane deviations for drowsy driver 14



(o) Duration of unmarked lane deviations for drowsy driver 15



(p) Duration of unmarked lane deviations for drowsy driver 16

Figure 2.13. Duration of unmarked lane deviations for the 16 drowsy drivers.

2.3 Settings for SVM Training and Testing

2.3.1 Preparation of Training Sets A and B

Like any SVM application, selecting appropriate vehicle variables as input variables was critical to the success of our SVM training. Guided by knowledge and intuition, we explored the following nine sets of SVM input variables: (1) lateral position, (2) lateral position and lateral velocity [defined as (current lateral position – previous lateral position) / T], (3) lateral position, lateral velocity, and speed, (4) lateral position, steering angle, and speed, (5) lateral position, steering angle, and yaw deviation, (6) lateral position, speed, and yaw deviation, (7) lateral position and yaw deviation, (8) lateral position and steering angle, and (9) lateral position, lateral velocity, and change in

steering angle. The speed was normalized to [0, 1] whereas the steering angle, yaw deviation, and change in steering angle were normalized to [-1, 1].

The prediction horizon for SVM 1 and SVM 2 (i.e., the amount of time in advance to detect a lane departure before it occurs) was set at three different levels: 0.2 s, 0.4 s or 0.6 s. Training examples were created from a lane departure event in an experiment by using the P data points prior to the lane departure moment. Each example consisted of N-point time series of the vehicle variables. Choosing appropriate time series length (i.e., time window size) is very important in order for the SVMs to attain the best prediction results. There is no general method to determine the optimal window size. In this study, the time series length for each prediction horizon was decided experimentally. Table 2.1 shows, as an illustrative instance, how 12 training examples are generated, where a lane departure is assumed to occur at time 17T. In this case, P=16, N=5, and the prediction horizon is 10 data points or 0.2 s (hence the lane departure class label for 7T to 16T is set 1). Each row of the five consecutive # symbols below the “Class Label” row represents a time series training example. Note that the class label of a training example is determined by the time instance of the last data point in the example (e.g., -1 for Example 1 and 1 for Example 3). The following relationship is obvious;

$$\text{number of training examples} = P \times \text{number of lane departure events} - N + 1$$

Each data point in the time series was treated as an input variable for the SVMs. Thus, there were 3N input variables if three vehicle variables were involved in the N-point long time series. For the lateral position and lateral velocity combination with 0.4 s and 0.6 s prediction horizon, every variable value was squared first before being fed to

the SVMs to enlarge the differences between the two consecutive values of the variables when the difference is larger (because $|a_2 - b_2| > |a - b|$ when $|a + b| > 1$). This operation was not applied to the 0.2 s prediction horizon because it was significantly easier for SVMs to classify the data with excellent results at this shorter horizon than they do at 0.4 s and 0.6 s. This operation was also applied to variable combinations (8) and (9).

We used all the odd number of lane departures (1,756 in total) as the base to create all the training examples for Training Set A. The total numbers of training examples for prediction horizon 0.2 s (using $P=36$ and $N=6$), 0.4 s (using $P=74$ and $N=8$), and 0.6 s (using $P=84$ and $N=10$) were 63,211, 129,937 and 147,495, respectively. They represented a total driving time of 21.07 min, 43.31 min and 49.17 min, respectively. Together, they represented less than 5% of the total driving time for the 22 drivers, which was 42 hours and 7 minutes (this is after the low speed data were removed).

Table 2.1. Illustration of constructing training examples from one lane departure event.

| | Time | | | | | | | | | | | | | | | | |
|-------------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| | T | $2T$ | $3T$ | $4T$ | $5T$ | $6T$ | $7T$ | $8T$ | $9T$ | $10T$ | $11T$ | $12T$ | $13T$ | $14T$ | $15T$ | $16T$ | $17T$ |
| Class Label | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Example 1 | # | # | # | # | # | | | | | | | | | | | | |
| Example 2 | | # | # | # | # | # | | | | | | | | | | | |
| Example 3 | | | # | # | # | # | # | | | | | | | | | | |
| : | | | | | | | | | | | | | | | | | |
| Example 12 | | | | | | | | | | | | # | # | # | # | # | |

To prepare the examples for Training Set B, we employed all the examples in Training Set A plus examples constructed using all the false positives (59 in total)

corresponding to Testing Set A in the case of 0.4 s prediction horizon and 80% of the false positives (150 in total) for 0.6 s prediction horizon. The number of training examples generated from a false positive event equals to the number of consecutive moments involving the data from Testing Set A at that time that were misclassified by SVM 1 as lane departure. As a result, the total numbers of training examples were 132,301 and 154,194, respectively, for the 0.4 s and 0.6 s prediction horizons. They represented a total driving time of 44.1 min and 51.4 min, respectively. Comparing to Training Set A, a total of 2,364 and 6,699 false positive examples were added to Training Sets B for the 0.4 s and 0.6 s prediction horizons, respectively. A Training Set B was not formed for the 0.2 s prediction horizon because there were only total two false positive cases for all the 22 drivers after the first-stage training, indicating the second-stage training is unnecessary.

2.3.2 Preparation of Testing Sets A and B

The trained SVM 1 was tested against the 16 drowsy drivers and 6 control drivers while the trained SVM 2 was assessed involving only the 16 drowsy drivers because there were too few false positives in the initial testing result to generate training examples.

The data in a Training Set A were excluded from the driver data files and each of the resulting driver files was then divided into two equal-length time series files from the middle point of time to form Testing Set A using the first resulting time series and Testing Set B using the second resulting time series. An N -point sliding time window moving one data point at a time was used to generate test cases from the Testing Sets for each driver ($N=6$, $N=8$, and $N=10$ for the prediction horizons 0.2 s, 0.4 s and 0.6 s, respectively, regardless of the driver type). The total numbers of testing examples for Testing Set A and

Testing Set B were the same. For prediction horizon 0.2 s (using $P=36$ and $N=6$), 0.4 s (using $P=74$ and $N=8$), and 0.6 s (using $P=84$ and $N=10$), they were 3,463,060, 3,429,697, and 3,420,918, respectively. They represented a total driving time of 19.239 hour, 19.054 hour, and 19.00 hour, respectively. A lane departure prediction was considered correct only if at least 4, 7, and 9 consecutive windows of a test case were classified by SVM 1 or SVM 2 as lane departure for the 0.2 s, 0.4 s, and 0.6 s prediction horizons, respectively.

2.3.3 Other SVM Settings

We experimented the linear, polynomial and RBF kernels for the SVMs, which are the most popular kernels in the literature. A 15-fold cross validation was first used to find good estimates for the initial values of the RBF kernel parameter σ and the SVM regularization parameter c . Then, different combinations of the c and σ values were explored experimentally to find the best combination. The same approach was used to find the best combination of the polynomial kernel parameter d and the regularization parameter c .

2.3.4 Statistical Measures of the Binary Classification Performance

Two statistical measures were used to evaluate the performance of our algorithm, sensitivity and specificity. The prediction result can be either one of two states; the driver is in the lane or the driver is about to drift off the lane. However, the results may or may not match the driver's actual status. Table 2.2 displays these states, where the rows show the true status of the driver while the columns indicate the classification results. The positive sign indicates that the driver was out of the lane while the negative sign means

the driver keeping the lane. Based on this table, sensitivity and specificity can be defined as following:

Sensitivity: is the ability of the algorithm to predict the lane deviations.

$$\text{Sensitivity} = \frac{a}{a+b} = \frac{\text{True positive}}{\text{True positive} + \text{false negative}} \quad (2.3)$$

Specificity: is the ability of the algorithm to identify the non-lane deviations.

$$\text{Specificity} = \frac{d}{d+c} = \frac{\text{True negative}}{\text{True negative} + \text{false positive}} \quad (2.4)$$

where,

True positive (a): drifting drivers correctly predicted as drifting drivers.

True negative (d): non-drifting drivers correctly classified as drivers still in the lane.

False positive (c): non-drifting drivers incorrectly identified as drifting drivers.

False negative (b): drifting drivers incorrectly classified as non-drifting drivers.

Table 2.2. Possible classification results of a driver [45].

| | | <i>Classification results</i> | |
|--------------------|------------------------|-------------------------------|--------------|
| | | Positive (+) | Negative (-) |
| <i>True status</i> | Lane deviation (+) | a | b |
| <i>Of a driver</i> | Non-Lane deviation (-) | c | d |

2.3.5 Statistical Pattern Recognition Toolbox for Matlab (STPRtool box)

We implemented the SVMs using the MATLAB-based freeware Statistical Pattern Recognition Toolbox (STPRtool) [44], which contained various pattern recognition methods, including the binary SVM. The toolbox was initially developed in 1999 and has been extended to include many pattern recognition algorithms. In our research we used the version 2.11 of the software, which was released in August 2011. Our SVM program ran on a PC with an i5 Intel CPU and 6 GB RAM. The computer execution time for the training ranged from 1 hour to 24.9 hours, depending on the input variables and settings. Table 2.3 shows some functions that was implemented in STPRtool box and the data type used to represent binary support vector machines is shown in Table 2.4.

Table 2.3. Implemented functions for binary support vector machines [44].

| Function | Description |
|-------------|--|
| svmclass | Support vector machine classifier. |
| evalsvm | Evaluates support vector machines classifier (for cross validation). |
| smo | Sequential minimal optimizer for binary SVMs with L1-soft margin. |
| svmlight | Interface to <i>SVMLight</i> software. |
| svmquadprog | SVM trained by Matlab optimization toolbox |

Table 2.4. Data type used to represent binary SVMs [44].

| Variable | Description |
|-------------------------------|---|
| .Alpha [$d \times 1$] | Weight vector α |
| .b [1×1] | Bias b. |
| .sv.X [$n \times d$] | Support vectors, s_1, s_2, \dots, s_d . |
| .options.ker [string] | Kernel identifier. |
| .options.arg [$1 \times p$] | Kernel argument. |
| .fun = 'svmclass' | Function identifier. |

The resulting SVM model includes several outputs, some of them listed below:

- *model.Alpha*: the optimal Lagrange multipliers obtained by solving the dual problem.
- *model.b*: the bias term in the decision function
- *model.nsv*: number of support vectors
- *model.trnerr*: the training error
- *model.margin*: the soft margin.
- *model.sv*: the support vectors
- *model.cputime*: time taken to build the model

CHAPTER 3 SVM LANE DEPARTURE PREDICTION EXPERIMENT RESULTS

3.1 SVM Training Results

Among the three kernel functions that we experimented, the linear kernel performed the worst - it failed to predict virtually any of the lane departures. This was to be expected because the time series leading to lane departure was not simple (e.g., the examples shown in Fig. 2.1). Moreover, among the nine input variable combinations, we found that the combination of lateral position and lateral velocity produced the best results for both SVM1 and SVM 2 as shown in Section 3.2. Hence, we will report only the training and testing results for the two SVMs that use the RBF kernel (with the experimentally determined optimal values of $c = 10$ and $\sigma = 0.1$) and the second-order polynomial kernel (with the experimentally determined optimal values of $c = 10$ and $d = 2$). Moreover, we will show only the training results involving lateral position and lateral velocity in the current subsection and will focus on this input combination in Section 3.2 that covers the testing results.

Table 3.1 shows how the false positives and false negatives change with different window sizes at 0.2 s, 0.4 s, and 0.6 s prediction horizons for SVM 1 when the RBF kernel was used and the input variables are lateral position and lateral velocity. Similar trends were observed for SVM 2 and the other eight input combinations, independent of the kernel types. As the window size increases, the number of false negatives increases and the number of false positives decreases. This is because for a shorter window size, the

SVMs become more sensitive to a quicker time series change (e.g., an abrupt lane departure) and thus is able to correctly predict lane departures more often. At the same time, however, small changes in the time series can be mistakenly treated as lane departures due to the higher sensitivity, leading to more false positives. For a longer window size, the SVMs have fewer false positives but at the same time tend to fail to recognize lane departures more frequently due to lower time sensitivity, which translates to more false negatives. The tradeoff between the numbers of false positives and false negatives led us to a balanced choice of the window sizes - 0.12 s, 0.16 s, and 0.20 s for 0.2 s, 0.4 s, and 0.6 s prediction horizons, respectively.

Table 3.1. How numbers of false positives and false negatives change with the window size for the RBF kernel SVM 1 using lateral position and lateral velocity as its inputs.

| Window size (s) | Number of false negatives | | | Number of false positives | | |
|-----------------|---------------------------|--------------------------|--------------------------|---------------------------|--------------------------|--------------------------|
| | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.6 s prediction horizon | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.6 s prediction horizon |
| 0.08 | 4 | 3 | 10 | 3 | 195 | 415 |
| 0.12 | 4 | 7 | 23 | 2 | 149 | 388 |
| 0.16 | 7 | 10 | 24 | 2 | 138 | 381 |
| 0.20 | 9 | 16 | 26 | 2 | 130 | 370 |
| 0.24 | 10 | 24 | 32 | 2 | 129 | 369 |

Table 3.2 presents the numbers of support vectors and training errors [defined as $\sum(\text{labels predicted by the classifier} \neq \text{true labels}) / \text{number of true labels}$] for SVM 1 and SVM 2 with different prediction horizons (0.2 s, 0.4 s, and 0.6s). For the RBF kernel SVM 1, the number of support vectors and the training error increase as the prediction

horizon increases. This is because the number of training data and the dimension (i.e., window size) of the data increase as the prediction horizon increases. For the same prediction horizon the number of support vectors with the RBF kernel is greater than that of the second-order polynomial kernel while the training error of the RBF kernel is always less than that of the second-order polynomial kernel. These reflect the ability of the kernels and the nature of this particular clarification problem. At the 0.4 s and 0.6 s prediction horizons, the second-order polynomial kernel SVM 1 correctly predicted only 23 and 54 of the 1775 lane departures, respectively, which are reflected by high amounts of training errors shown in Table 3.2. While the RBF kernel performed far better than the second-order polynomial kernel at the 0.4 s, and 0.6 s prediction horizons, its performance degraded markedly at the 0.6 s prediction horizon. The number of support vectors and the training error for SVM 2 are greater than the corresponding ones for SVM 1. This is because the number of training examples for SVM 2 is greater than that of SVM 1 for the same prediction horizon as the training data contains the misclassified examples of SVM 1, making it more difficult for SVM 2 to classify. The ratio of the training error to the number of examples for SVM 1 and SVM 2 at 0.4 s prediction horizon are 1.170×10^{-7} and 1.217×10^{-7} error/example, respectively. These ratios increased to 1.783×10^{-7} and 2.250×10^{-7} at 0.6 s prediction horizon, respectively, as the classification task became more challenging.

Table 3.2. Amounts of support vectors and training error for SVM 1 using Training Set A and SVM 2 using Training Set B. Lateral position and lateral velocity are input variables.

| Kernel function | Number of support vectors | | | Training error | | |
|--|---------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.6 s prediction horizon | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.6 s prediction horizon |
| RBF (SVM 1) | 5251 | 12452 | 18765 | 0.0059 | 0.0152 | 0.0263 |
| Second-order polynomial (SVM 1) | 2247 | 4877 | 8780 | 0.0071 | 0.534 | 0.466 |
| RBF (SVM 2) | NA | 13835 | 20293 | NA | 0.0161 | 0.0347 |

3.2 SVM Testing Results

To determine which input variable combination yields the best outcome, Table 3.3 shows the numbers of false negatives and false positives for all the nine input variable combinations at 0.2 s and 0.4 s prediction horizons when either the RBF or second-order polynomial kernel is used by SVM 1. As the prediction horizon becomes longer, the number of false positives and false negatives get worse. Moreover, when one input variable combination performs better than the rest of the combinations at 0.2 s prediction horizon, it most likely does better at 0.4 s prediction horizon. These observed patterns extend to 0.6 s prediction horizon, which are not shown here. At the 0.2 s prediction horizon with RBF kernel, input variable combinations 1, 2, and 8 have the fewest false negatives (which are 0, 4, and 3, respectively). Among the three combinations, combination 2 has the least false positives, which is 2 only. Therefore, if the false positives and false negatives must be counted at the same time, combination 2 (i.e., lateral position and lateral velocity) stands out as the best combination among the nine. Note that even though combination 9 has only 1 false positive, which is less than 2 false

positives of combination 2, it has 7 false negatives, much higher than 4 false negatives of combination 2. Reasoning along this line leads to the conclusions that combination 2 is the best for the RBF kernel at the 0.4 s prediction horizon and for the second-order polynomial kernel at the 0.2 s prediction horizon. Note that at 0.4 s prediction horizon, the second-order polynomial SVM 1 performed much worse than the RBF SVM 1 and did poorly in general for every input variable combination. Due to this fact as well as the desire of having a SVM with as long a prediction horizon as possible, we will focus on below the testing results of (1) second-order polynomial SVM 1 at 0.2 s prediction horizon and input variable combination 2, (2) SVM 1 and SVM 2 involving the RBF kernel and input variable combination 2 at 0.2 s, 0.4 s, and 0.6 s prediction horizons. It is

Table 3.3. Numbers of false negatives and false positives generated during the testing of the RBF kernel SVM 1 and the second-order polynomial kernel SVM 1.

| | Input variable combination | Number of false negatives (RBF/polynomial) | | Number of false positives (RBF/polynomial) | | Total number of lane departures correctly predicted (RBF/polynomial) | |
|---|--|--|--------------------------|--|--------------------------|--|--------------------------|
| | | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.2 s prediction horizon | 0.4 s prediction horizon |
| 1 | Lateral position | 0/0 | 8/1775 | 73/54 | 342/0 | 1775/1775 | 1767/0 |
| 2 | Lateral position and lateral velocity | 4/0 | 10/1752 | 2/0 | 138/3657 | 1771/1775 | 1765/23 |
| 3 | Lateral position, lateral velocity, and speed | 18/19 | 33/1337 | 13/0 | 201/6 | 1757/1756 | 1742/438 |
| 4 | Lateral position, steering angle, and speed | 20/22 | 36/1737 | 275/299 | 829/4578 | 1755/1753 | 1739/38 |
| 5 | Lateral position, steering angle, and yaw deviation | 75/2 | 93/1219 | 389/35 | 967/1495 | 1700/1773 | 1682/556 |
| 6 | Lateral position, speed, and yaw deviation | 74/13 | 94/793 | 262/261 | 739/5612 | 1701/1762 | 1681/982 |
| 7 | Lateral position and yaw deviation | 8/1 | 22/1197 | 112/51 | 462/1607 | 1767/1774 | 1753/578 |
| 8 | Lateral position and steering angle | 3/2 | 3/496 | 132/33 | 333/1548 | 1773/1773 | 1773/1279 |
| 9 | Lateral position, lateral velocity, and change in steering angle | 7/0 | 14/20 | 1/1 | 193/1512 | 1768/1775 | 1761/1755 |

important to point out that unlike some of the vehicle variables that can be measured in simulated driving only, lateral position and lateral velocity can luckily be obtained in practice during vehicle operation.

Table 3.4, Table 3.5, and Table 3.6 show the testing results of RBF SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.2 s, 0.4 s, and 0.6 s, respectively. At the 0.2 s prediction horizon, there is no false positive or false negative for the 6 control drivers, and only 2 false positives and 4 false negatives for the 16 drowsy drivers, which translates to only one false positive alarm for approximately every 18.5 hours of driving. The overall sensitivity and specificity of the SVM with the 0.2 s prediction horizon are 99.77465% and 99.99997%, respectively. Accuracy for the 0.4 s prediction horizon is substantially lower –137 false positives and 10 false negatives for the 16 drowsy drivers and 1 false positive and 0 false negative for the 6 control drivers. The overall sensitivity and specificity for the 22 drivers are 99.43662% and 99.99799%, respectively. As the prediction horizon became longer, the prediction accuracy got worsen, and the overall sensitivity and specificity at the 0.6 s prediction horizon for the 22 drivers decreased quite significantly to 98.53521% and 99.99454%, respectively. Table 3.7 presents the overall averages, the averages of average prediction horizon of drivers, and standard deviations, the standard deviations of averages prediction horizons of drivers, of the actual prediction horizons for the 22 drivers. The averages are very close to the intended prediction horizons. Furthermore, the standard deviations were very small, indicating the intended prediction horizons were achieved evenly by the drivers.

Table 3.4. Testing results of RBF SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.2 s prediction horizons.

| Driver (c – control; d – drowsy) | Number of lane departures | Number of lane departures correctly predicted | Number of lane departures failed to be predicted | Number of falsely predicted lane departures | Number of non-lane departures correctly identified |
|---|--|--|---|--|---|
| c1 | 0 | 0 | 0 | 0 | 62831 |
| c2 | 6 | 6 | 0 | 0 | 60785 |
| c3 | 2 | 2 | 0 | 0 | 64492 |
| c4 | 0 | 0 | 0 | 0 | 63927 |
| c5 | 8 | 8 | 0 | 0 | 62749 |
| c6 | 1 | 1 | 0 | 0 | 51535 |
| d1 | 25 | 25 | 0 | 0 | 384500 |
| d2 | 131 | 131 | 0 | 0 | 402594 |
| d3 | 29 | 29 | 0 | 0 | 484227 |
| d4 | 74 | 73 | 1 | 0 | 410394 |
| d5 | 170 | 170 | 0 | 0 | 400528 |
| d6 | 177 | 177 | 0 | 0 | 365205 |
| d7 | 52 | 52 | 0 | 0 | 459118 |
| d8 | 132 | 132 | 0 | 0 | 412464 |
| d9 | 108 | 108 | 0 | 0 | 421482 |
| d10 | 105 | 105 | 0 | 0 | 437095 |
| d11 | 130 | 130 | 0 | 0 | 412868 |
| d12 | 43 | 43 | 0 | 0 | 434074 |
| d13 | 97 | 97 | 0 | 0 | 433421 |
| d14 | 195 | 194 | 1 | 1 | 338774 |
| d15 | 110 | 109 | 1 | 0 | 398396 |
| d16 | 180 | 179 | 1 | 1 | 346529 |
| Total | 1775 | 1771 | 4 | 2 | 6907988 |

Table 3.5. Testing results of RBF SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.4 s prediction horizons.

| Driver (c – control; d – drowsy) | Number of lane departures | Number of lane departures correctly predicted | Number of lane departures failed to be predicted | Number of falsely predicted lane departures | Number of non-lane departures correctly identified |
|---|--|--|---|--|---|
| c1 | 0 | 0 | 0 | 0 | 62829 |
| c2 | 6 | 6 | 0 | 0 | 60644 |
| c3 | 2 | 2 | 0 | 0 | 64467 |
| c4 | 0 | 0 | 0 | 0 | 63925 |
| c5 | 8 | 8 | 0 | 0 | 62643 |
| c6 | 1 | 1 | 0 | 1 | 51497 |
| d1 | 25 | 25 | 0 | 0 | 383941 |
| d2 | 131 | 131 | 0 | 8 | 399642 |
| d3 | 29 | 29 | 0 | 7 | 483459 |
| d4 | 74 | 72 | 2 | 7 | 408729 |
| d5 | 170 | 170 | 0 | 10 | 396567 |
| d6 | 177 | 176 | 1 | 6 | 361312 |
| d7 | 52 | 52 | 0 | 10 | 457872 |
| d8 | 132 | 132 | 0 | 8 | 409414 |
| d9 | 108 | 108 | 0 | 15 | 418900 |
| d10 | 105 | 105 | 0 | 10 | 434688 |
| d11 | 130 | 130 | 0 | 9 | 409908 |
| d12 | 43 | 43 | 0 | 2 | 433133 |
| d13 | 97 | 97 | 0 | 8 | 431157 |
| d14 | 195 | 193 | 2 | 13 | 334420 |
| d15 | 110 | 108 | 2 | 8 | 395914 |
| d16 | 180 | 177 | 3 | 16 | 342441 |
| Total | 1775 | 1765 | 10 | 138 | 6867502 |

Table 3.6. Testing results of RBF SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.6 s prediction horizons.

| Driver (c – control; d – drowsy) | Number of lane departures | Number of lane departures correctly predicted | Number of lane departures failed to be predicted | Number of falsely predicted lane departures | Number of non-lane departures correctly identified |
|---|--|--|---|--|---|
| c1 | 0 | 0 | 0 | 0 | 62827 |
| c2 | 6 | 6 | 0 | 0 | 54691 |
| c3 | 2 | 2 | 0 | 0 | 64347 |
| c4 | 0 | 0 | 0 | 0 | 63923 |
| c5 | 8 | 8 | 0 | 0 | 62158 |
| c6 | 1 | 1 | 0 | 1 | 51409 |
| d1 | 25 | 24 | 1 | 9 | 382606 |
| d2 | 131 | 130 | 1 | 18 | 393065 |
| d3 | 29 | 29 | 0 | 10 | 481828 |
| d4 | 74 | 73 | 1 | 11 | 405069 |
| d5 | 170 | 169 | 1 | 33 | 387919 |
| d6 | 177 | 172 | 5 | 20 | 352851 |
| d7 | 52 | 52 | 0 | 22 | 454866 |
| d8 | 132 | 132 | 0 | 23 | 402819 |
| d9 | 108 | 107 | 1 | 36 | 413013 |
| d10 | 105 | 104 | 1 | 30 | 429061 |
| d11 | 130 | 129 | 1 | 31 | 403274 |
| d12 | 43 | 43 | 0 | 9 | 430915 |
| d13 | 97 | 97 | 0 | 26 | 426133 |
| d14 | 195 | 189 | 6 | 33 | 324717 |
| d15 | 110 | 106 | 4 | 21 | 390238 |
| d16 | 180 | 176 | 4 | 37 | 333412 |
| Total | 1775 | 1749 | 26 | 370 | 6771141 |

Table 3.7. Overall averages and standard deviations of the actual prediction horizons of RBF SVM 1.

| | Overall average of the actual prediction horizons (s) | | | Standard deviation of the actual prediction horizons (s) | | |
|--------------------------|---|--------------------------|--------------------------|--|--------------------------|--------------------------|
| | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.6 s prediction horizon | 0.2 s prediction horizon | 0.4 s prediction horizon | 0.6 s prediction horizon |
| 6 control drivers | 0.1987 | 0.4042 | 0.5995 | 0.0063 | 0.0107 | 0.0107 |
| 16 drowsy drivers | 0.2005 | 0.3996 | 0.5922 | 0.0021 | 0.0078 | 0.0078 |

Table 3.8 shows the testing results of the SVM trained to predict lane departure 0.2 s in advance using the second-order polynomial kernel as at the 0.4s, and 0.6 s prediction horizons the SVM 1 with second-order polynomial kernel failed to predict most of the lane departures. The overall average of the actual prediction horizon at 0.2 s prediction horizon is 0.200728 s. The overall sensitivity and specificity of the SVM with the 0.2-second prediction horizon for the 22 drivers are 100% (there are no false positive or negative for the control and drowsy drivers).

Table 3.8. Testing results of second-order polynomial SVM 1 using Testing Sets A and B with lateral position and lateral velocity being input variables at 0.2s prediction horizons.

| Driver (c – control; d – drowsy) | Number of lane departures | Number of lane departures correctly predicted | Number of lane departures failed to be predicted | Number of falsely predicted lane departures | Number of non-lane departures correctly identified |
|---|--|--|---|--|---|
| c1 | 0 | 0 | 0 | 0 | 62831 |
| c2 | 6 | 6 | 0 | 0 | 60785 |
| c3 | 2 | 2 | 0 | 0 | 64492 |
| c4 | 0 | 0 | 0 | 0 | 63675 |
| c5 | 8 | 8 | 0 | 0 | 62747 |
| c6 | 1 | 1 | 0 | 0 | 51535 |
| d1 | 25 | 25 | 0 | 0 | 384500 |
| d2 | 131 | 131 | 0 | 0 | 402591 |
| d3 | 29 | 29 | 0 | 0 | 484227 |
| d4 | 74 | 74 | 0 | 0 | 410391 |
| d5 | 170 | 170 | 0 | 0 | 400519 |
| d6 | 177 | 177 | 0 | 0 | 365209 |
| d7 | 52 | 52 | 0 | 0 | 459116 |
| d8 | 132 | 132 | 0 | 0 | 412459 |
| d9 | 108 | 108 | 0 | 0 | 421479 |
| d10 | 105 | 105 | 0 | 0 | 437093 |
| d11 | 130 | 130 | 0 | 0 | 412868 |
| d12 | 43 | 43 | 0 | 0 | 434069 |
| d13 | 97 | 97 | 0 | 0 | 433411 |
| d14 | 195 | 195 | 0 | 0 | 338775 |
| d15 | 110 | 110 | 0 | 0 | 398395 |
| d16 | 180 | 180 | 0 | 0 | 346539 |
| Total | 1775 | 1775 | 0 | 0 | 6907706 |

The SVM 2 was tested against Testing Set B. The main goal of SVM 2 is to reduce false positives. Hence, it was not tested against the 6 control drivers because they produced only one false positive at 0.4 s and 0.6 s prediction horizons during the testing of SVM 1. For the same reason, SVM 2 was not tested against any control or drowsy driver at the 0.2 s prediction horizon because the 22 drivers together generated only 2 false positives at this horizon for RBF kernel and zero false positive for the second-order polynomial kernel. To show the benefit of the second-stage training, Tables 3.9 and 3.10 compares the testing results of SVM 1 and SVM 2, both of which used the same kernel (i.e., the RBF) and the same input variables (i.e., lateral position and lateral velocity) and were tested against the 16 drowsy drivers in Testing Set B at 0.4 s and 0.6 s, respectively. The ability of SVM 2 to identify lane departure was slightly better than that of SVM 1. However, SVM 2 generated far less false positives than SVM 1 (18 vs. 78 for the 0.4 s prediction horizon and 63 vs. 181 for the 0.6 s prediction horizon). The numbers of false negatives for SVM 1 and SVM 2 were 7 and 3, respectively, at 0.4 prediction horizon and 11 and 9 at 0.6 s prediction horizon. Even though the numbers of true positives and the numbers of false positives for SVM 1 and SVM 2 are the same in some cases, numbers of true negatives can be different. The number of examples in a true or false lane departures are different. A lane departure prediction is considered only if at least 7, and 9 consecutive examples of a test case are classified by SVM 1 or SVM 2 as lane departure for the 0.4 s, and 0.6 s prediction horizons, respectively. If less than 7 and 9 consecutive examples of a test case are mistakenly classified by SVM 1 as lane departures for 0.4 s and 0.6 s prediction horizons, they will not be counted as lane departures. However, if

they classified correctly as non-lane departures by SVM 2, this will affect only the number of true negatives. A true non-lane departure is counted for any single window of a test case is classified by SVM 1 or SVM 2 as a non-lane departure. At the 0.4 s prediction horizon, the overall sensitivity and specificity for SVM 1 are 99.33205% and 99.99759%, respectively. The corresponding measures for SVM 2 are improved to 99.71374% and 99.99944%, respectively. At the 0.6 s prediction horizons, the overall sensitivity and specificity for SVM 1 are 98.95038% and 99.99438%, respectively, and 99.14122% and 99.99805%, respectively, for SVM 2, which shows the same improving trends. All these comparison data demonstrate the effectiveness of the second-stage training, resulting in a better SVM 2 than SVM 1.

Table 3.11 depicts the overall averages and standard deviations of the actual prediction horizons achieved by SVM 1 and SVM 2 when tested against the 16 drowsy drivers in Testing Set B. The actual prediction horizons are very close to the expected horizons for both SVMs. The standard deviations were very small, meaning the actual prediction horizons were attained by all the drivers in a uniform fashion.

Table 3.9. Testing results of the RBF SVM 1 and the RBF SVM2 using the 16 drowsy drivers in Testing Set B with lateral position and lateral velocity as input variables at 0.4s prediction horizons.

| Driver | Number of lane departure | Number of lane departure correctly predicted (SVM 1/SVM 2) | Number of falsely predicted lane departure (SVM 1/SVM 2) | Number of non-lane departure correctly identified (SVM 1/SVM 2) |
|---------------|---------------------------------|---|---|--|
| d1 | 7 | 7/7 | 0/0 | 192416/192404 |
| d2 | 95 | 95/95 | 4/0 | 197926/198034 |
| d3 | 18 | 18/18 | 2/0 | 241533/241583 |
| d4 | 63 | 61/63 | 7/3 | 202609/202708 |
| d5 | 102 | 102/102 | 4/2 | 197166/197282 |
| d6 | 109 | 109/109 | 3/2 | 179334/179420 |
| d7 | 34 | 34/34 | 5/0 | 228435/228525 |
| d8 | 74 | 74/74 | 5/1 | 204165/204326 |
| d9 | 69 | 69/69 | 13/5 | 208357/208598 |
| d10 | 58 | 58/58 | 4/0 | 217063/217161 |
| d11 | 83 | 83/83 | 5/1 | 203731/203873 |
| d12 | 19 | 19/19 | 2/0 | 216757/216790 |
| d13 | 47 | 47/47 | 4/0 | 215687/215776 |
| d14 | 89 | 87/88 | 6/0 | 167785/167975 |
| d15 | 78 | 76/77 | 2/1 | 196466/196512 |
| d16 | 103 | 102/102 | 12/3 | 170307/170493 |
| Total | 1048 | 1041/1045 | 78/18 | 3239737/3241460 |

Table 3.10. Testing results of the RBF SVM 1 and the RBF SVM2 using the 16 drowsy drivers in Testing Set B with lateral position and lateral velocity as input variables at 0.6s prediction horizons.

| Driver | Number of lane departure | Number of lane departure correctly predicted (SVM 1/SVM 2) | Number of falsely predicted lane departure (SVM 1/SVM 2) | Number of non-lane departure correctly identified (SVM 1/SVM 2) |
|---------------|---------------------------------|---|---|--|
| d1 | 7 | 7/7 | 3/1 | 192265/192323 |
| d2 | 95 | 95/95 | 6/2 | 196705/197023 |
| d3 | 18 | 18/18 | 4/2 | 241176/241327 |
| d4 | 63 | 62/63 | 11/4 | 201654/201990 |
| d5 | 102 | 102/102 | 14/8 | 195721/196031 |
| d6 | 109 | 108/108 | 8/5 | 178031/178219 |
| d7 | 34 | 34/34 | 14/5 | 227746/228027 |
| d8 | 74 | 74/74 | 11/7 | 203148/203426 |
| d9 | 69 | 68/68 | 22/7 | 207126/207736 |
| d10 | 58 | 58/58 | 12/0 | 216080/216510 |
| d11 | 83 | 82/83 | 16/6 | 202560/202866 |
| d12 | 19 | 19/19 | 7/0 | 216382/216583 |
| d13 | 47 | 47/47 | 10/2 | 214926/215207 |
| d14 | 89 | 87/87 | 16/7 | 166383/166896 |
| d15 | 78 | 75/75 | 8/2 | 195410/195668 |
| d16 | 103 | 101/101 | 19/5 | 168848/169348 |
| Total | 1048 | 1037/1039 | 181/63 | 3224161/3229180 |

Table 3.11. Overall averages and standard deviations of the actual prediction horizons of SVM 1 and SVM 2 tested against the 16 drowsy drivers in Testing Set B.

| | Overall average of the actual prediction horizons (s) | | Standard deviation of the actual prediction horizons (s) | |
|--------------|---|--------------------------|--|--------------------------|
| | 0.4 s prediction horizon | 0.6 s prediction horizon | 0.4 s prediction horizon | 0.6 s prediction horizon |
| SVM 1 | 0.39833 | 0.59001 | 0.00826 | 0.01521 |
| SVM 2 | 0.39617 | 0.59013 | 0.00664 | 0.01055 |

CHAPTER 4 CONCLUSION

We explored the nonlinear binary two-stage SVM training scheme with different kernel functions to predict lane departure using time series of the vehicle variables with different prediction horizons. This SVM's predictive ability was experimentally optimized by finding the best SVM kernel and its parameter values and the most appropriate set of vehicle variables (among nine variable sets), which turned out to be the lateral position and lateral velocity. It was found that the SVM 1 with the second-order polynomial kernel provided the best outcome for 0.2 s prediction horizon, but the results became very bad when the prediction horizon was longer. The SVM 1 with the radial basis function kernel was the only kernel (among the kernels that we tested) that produced the best results for 0.4 s and 0.6 s prediction horizons. The SVM's prediction ability at 0.4 s and 0.6 s prediction horizons with the radial basis function kernel was improved by using the SVM 2. The testing results showed that SVM 2 performed much better than SVM 1 in terms of false positives. Testing results involving 16 drowsy drivers and 6 control drivers with a total of over 6.84 million prediction decisions (reflecting more than 38 hours of total driving time) demonstrated significant SVM performance.

The target of the future work is to improve the prediction horizon of the developed LDWS (i.e. try 0.8 s and 1 s prediction horizons). The improvement includes reducing the false positives (false alarms) and the false negatives, which means increasing the capability of the algorithm to predict the real lane deviations. Some of the steps that may be applied in order to get better results include:

1. Explore more input variable combinations. As we mentioned before, features selection has an enormous impact on the success of learning algorithms to recognize complex patterns and make intelligent decisions based on data. It is very important to select the most effective features rather than all available features. Selecting the best combination of variables sometimes requires a systematic approach; however, the ideal way to select the best features is to experiment all possible combination of features as input to the learning machine. By a common sense, the lateral position of the vehicle should be included in all possible input subsets, that because the lateral position demonstrated the most consistent pattern among all the other features.

2. Use different kernel functions. No doubt that the choices of the kernel function is a crucial for generalization capabilities of the learning techniques. However, up to now there is no a general method to find a best kernel function. We tested three types of kernels with different combinations of features to find the kernel function that leads to best performance. The tested kernels were linear, polynomial , and RBF kernel. Actually, there are some other types of kernels need to be tested. The possible kernels that can be used include hyperbolic tangent (Sigmoid) Kernel and exponential Kernel. Moreover, different kernel's parameters need to be experimented in order to optimize the error performance of the classifier.

REFERENCES

- [1] L. T. Lam, "Distractions and the risk of car crash injury: The effect of drivers' age," *Journal of Safety Research*, vol. 33, pp. 411-419, 2002.
- [2] D. A. Redelmeier and R. J. Tibshirani, "Association between cellular-telephone calls and motor vehicle collisions," *New England Journal of Medicine*, vol. 336, pp. 453-458, 1997.
- [3] T.-H. Chang, C.-S. Hsu, C. Wang, and L.-K. Yang, "Onboard measurement and warning module for irregular vehicle behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 501-513, 2008.
- [4] U. S. NHTSA. (2009, 1/20/2012). "Traffic Safety Facts". Available:
<http://www.nrd.nhtsa.dot.gov/Pubs/811172.pdf>
- [5] M. Tideman, M. C. van der Voort, B. van Arem, and F. Tillema, "A Review of Lateral Driver Support Systems," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, 2007, pp. 992-999.
- [6] A. Houser, J. Pierowicz, D. Fuglewicz, U. S. F. M. C. S. Administration, and C. Corporation, *Concept of operations and voluntary operational requirements for lane departure warning systems (LDWS) on-board commercial motor vehicles*: US Department of Transportation, Federal Motor Carrier Safety Administration, 2005.

- [7] H. Pei-Yung and Y. Chun-Wei, "A Portable Real-Time Lane Departure Warning System based on Embedded Calculating Technique," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, 2006, pp. 2982-2986.
- [8] NHTSA. (2010). *National Center for Statistics and Analysis. Traffic Safety Facts 2010. U.S. Department of Transportations, National Highway Traffic Safety Administration*. Available: <http://www-nrd.nhtsa.dot.gov/Pubs/811552.pdf>
- [9] A. R. Ahmad, M. Khalid, and R. Yusof, "Machine Learning Using Support Vector Machines," *Centre for Artificial Intelligence and Robotics*.
- [10] H. Godthelp, P. Milgram, and G. J. Blaauw, "The development of a time-related measure to describe driving strategy," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 26, pp. 257-268, 1984.
- [11] L. Chiu-Feng and A. G. Ulsoy, "Calculation of the time to lane crossing and analysis of its frequency distribution," in *American Control Conference, 1995. Proceedings of the*, 1995, pp. 3571-3575 vol.5.
- [12] C. Mei, T. Jochem, and D. Pomerleau, "AURORA: a vision-based roadway departure warning system," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, 1995, pp. 243-248 vol.1.
- [13] G. Cario, A. Casavola, G. Franz'e, M. Lupia, G. Brasili, and I. it SpA, "Predictive Time-To-Lane-Crossing Estimation For Lane Departure Warning Systems," ed: U. S. Department of Transportation, 1200 New Jersey Avenue SE Washington DC 20590 USA, 2009.

- [14] P. Angkititrakul, R. Terashima, and T. Wakita, "On the Use of Stochastic Driver Behavior Model in Lane Departure Warning," *Intelligent Transportation Systems, IEEE Transactions on*, pp. 1-10, 2011.
- [15] W. Van Winsum, K. Brookhuis, and D. De Waard, "A comparison of different ways to approximate time-to-line crossing (TLC) during car driving," *Accident Analysis & Prevention*, vol. 32, pp. 47-56, 2000.
- [16] S. Mammar, S. Glaser, and M. Netto, "Time to line crossing for lane departure avoidance: A theoretical study and an experimental setting," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, pp. 226-241, 2006.
- [17] L. Sukhan, K. Woong, and L. Jae-Won, "A vision based lane departure warning system," in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, 1999, pp. 160-165 vol.1.
- [18] A. Gern, R. Moebus, and U. Franke, "Vision-based lane recognition under adverse weather conditions using optical flow," in *Intelligent Vehicle Symposium, 2002. IEEE*, 2002, pp. 652-657 vol.2.
- [19] J. W. Lee, "A machine vision system for lane-departure detection," *Computer vision and image understanding*, vol. 86, pp. 52-78, 2002.
- [20] H. Pau-Lo, C. Hsu-Yuan, T. Bol-Yi, and H. Wen-Jing, "The adaptive lane-departure warning system," in *SICE 2002. Proceedings of the 41st SICE Annual Conference*, 2002, pp. 2867-2872 vol.5.

- [21] C. D'Cruz and Z. Jua Jia, "Lane detection for driver assistance and intelligent vehicle applications," in *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*, 2007, pp. 1291-1296.
- [22] S. P. Tseng, L. Yung-Sheng, Y. Chih-Hsie, and H. Li-Kung, "A DSP-based lane recognition method for the lane departure warning system of smart vehicles," in *Networking, Sensing and Control, 2009. ICNSC '09. International Conference on*, 2009, pp. 823-828.
- [23] Z. Shengyan, J. Yanhua, X. Junqiang, G. Jianwei, X. Guangming, and C. Huiyan, "A novel lane detection based on geometrical model and Gabor filter," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 59-64.
- [24] L. Yu-Chi and C. Chieh-Li, "Vision-based lane departure detection system in urban traffic scenes," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, 2010, pp. 1875-1880.
- [25] C. Qingyang, S. Zhenping, C. Tongtong, and C. Liu, "Prediction of unintended lane departure based on detection of lane boundary," in *Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference on*, 2011, pp. 65-70.
- [26] L. Jing-Fu, S. Yi-Feng, K. Ming-Kuan, and Y. Pen-Ning, "Development of a Vision-Based Driver Assistance System with Lane Departure Warning and Forward Collision Warning Functions," in *Computing: Techniques and Applications, 2008. DICTA '08. Digital Image*, 2008, pp. 480-485.

- [27] V. Gaikwad and S. Lokhande, "An improved lane departure method for Advanced Driver Assistance System," in *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, 2012, pp. 1-5.
- [28] D. O. Cualain, C. Hughes, M. Glavin, and E. Jones, "Automotive standards-grade lane departure warning system," *Intelligent Transport Systems, IET*, vol. 6, pp. 44-57, 2012.
- [29] P. H. Batavia, "Driver-adaptive lane departure warning systems," Carnegie Mellon University, 1999.
- [30] R. Risack, N. Mohler, and W. Enkelmann, "A video-based lane keeping assistant," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 356-361.
- [31] C. Tang-Hsien, H. Chih-Sheng, W. Chieh, and Y. Li-Kai, "Onboard Measurement and Warning Module for Irregular Vehicle Behavior," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, pp. 501-513, 2008.
- [32] D. Xun, A. Kummert, P. Su Birm, and D. Neisius, "A warning algorithm for Lane Departure Warning system," in *Intelligent Vehicles Symposium, 2009 IEEE*, 2009, pp. 431-435.
- [33] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, pp. 121-167, 1998.
- [34] C. Campbell. (2008). *Introduction to Support Vector Machines* [Video lecture]. Available: <http://videolectures.net>

- [35] S. Theodoridis and K. Koutroumbas, "*Pattern Recognition*", Fourth ed.: Elsevier, 2009.
- [36] J. Shawe-Taylor and N. Cristianini, "Support Vector Machines," ed: Cambridge University Press, 2000.
- [37] C. R. Souza. (2010). "*Kernel Functions for Machine Learning Applications*". Available: <http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html>.
- [38] T. Evgeniou, M. Pontil, and T. Poggio, "Statistical learning theory: A primer," *International Journal of Computer Vision*, vol. 38, pp. 9-13, 2000.
- [39] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *The Journal of Machine Learning Research*, vol. 5, pp. 27-72, 2004.
- [40] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Data preprocessing for supervised learning," *International Journal of Computer Science*, vol. 1, pp. 111-117, 2006.
- [41] K. Kozak, J. Pohl, W. Birk, J. Greenberg, B. Artz, M. Blommer, L. Cathey, and R. Curry, "Evaluation of lane departure warnings for drowsy drivers," 2006, pp. 2400-2404.
- [42] H. Kozak, B. Artz, M. Blommer, L. Cathey, R. Curry, and J. Greenberg, "Evaluation of HMI for Lane departure Warning systems for Drowsy Drivers: A VIRTTEX Simulator Study," ed: Ford Motor Company, 2005.

- [43] A. Eskandarian, R. Sayed, P. Delaigue, A. Mortazavi, and J. Blum, "Advanced Driver Fatigue Research," United States 2007.
- [44] V. Franc and V. Hlavác, "Statistical pattern recognition toolbox for Matlab," *Center for Machine Perception, Czech Technical University*, 2004.
- [45] R. Parikh, A. Mathai, S. Parikh, G. C. Sekhar, and R. Thomas, "Understanding and using sensitivity, specificity and predictive values," *Indian Journal of Ophthalmology*, vol. 56, p. 45, 2008.

ABSTRACT**VEHICLE LANE DEPARTURE PREDICTION BASED ON
SUPPORT VECTOR MACHINES**

by

ALHADI ALI ALBOUSEFI

August 2014

Advisor: Prof. Hao Ying**Major:** Electrical Engineering**Degree:** Doctor of Philosophy

Advanced driver assistance systems, such as unintentional lane departure warning systems, have recently drawn much attention and R & D efforts. Such a system will assist the driver by monitoring the driver or vehicle behaviors to predict/detect driving situations (e.g., lane departure) and alert the driver to take corrective action. In this dissertation, we explored utilizing the nonlinear binary support vector machine (SVM) technique and the time series of vehicle variables to predict unintentional lane departure, which is innovative as no machine learning technique has previously been attempted for this purpose in the literature. Furthermore, we developed a two-stage training scheme to improve SVM's prediction performance. Our SVMs were trained and tested using the experiment data generated by VIRTTEX, a hydraulically powered 6-degrees-of-freedom moving base driving simulator at Ford Motor Company. The data represented 16 drowsy drivers (about three-hour driving time per subject) and six control

drivers (approximately 20 minutes driving per subject), all of which drove a simulated 2000 Volvo S80. More than 100 vehicle variables were sampled at 50 Hz. There were a total of 3,508 unintentional lane departure occurrences for the 16 drowsy drivers and 23 for four of the six control drivers (two had none). We optimized the performances of the SVMs by experimentally finding their best kernel functions and parameter values as well as the most appropriate vehicle variables as their input variables. Our experiment results involving the 22 drivers with a total of over 6.84 million prediction decisions demonstrate that: (1) the two-stage training scheme significantly outperformed the commonly used (one-stage) training scheme, (2) excellent SVM performances, as measured by numbers of false positives and false negatives, were achieved when the prediction horizon was set at 0.6 s or shorter, (3) lateral position and lateral velocity served as the best input variables among the nine variable sets that we explored, and (4) the radical basis function was the best kernel function (the other two kernel functions that we tested were the linear function and the second-order polynomial). We conclude that the two-stage-training SVM approach deserves further exploration because to the best of our knowledge, it has demonstrated the best unintentional lane departure prediction performance relative to the literature.

AUTOBIOGRAPHICAL STATEMENT

ALHADI ALI ALBOUSEFI

I was born in Surman, Libya. I finished my high school from Surman High School in Surman, Libya. I received BS in Electrical Engineering in 1993 from Tripoli University, Tripoli, Libya. To further my education, I transferred to Budapest University of Technology and Economics, Budapest, Hungary, where I received MS in Electrical Engineering in 2005, in honor standing.